84K00210
April 29, 1997

Basic

Checkout and Launch Control System (CLCS) System Design Document

Prepared By:

CLCS System Specification Team
CLCS Technical Integration
DE-CLC
Kennedy Space Center, FL 32899

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. Introduction

1.1 SCOPE

This document describes the system architecture of the Checkout and Launch Control System (CLCS). It discusses the conceptual model, data flow, interfaces and functions of the CLCS. In addition, subsystem functional allocation and hardware and computer software configuration items are identified.

1.2 System Overview

The CLCS is composed of the Real Time Processing System (RTPS), the Shuttle Data Center (SDC), the Simulation System (SIM), and the Business Information Network (BIN). The RTPS provides the capability to monitor and control the elements of the current Space Shuttle Flight Vehicle and Ground Support Equipment (GSE). The SDC is the repository for the Shuttle Launch Processing Test Data and provides the capability to build Test Packages for configuration of the RTPS. The capability to debug and certify RTPS software and to aid in the training of checkout and launch personnel is provided by the SIM. The BIN provides RTPS workstation connectivity and access to non-RTPS applications and data. CLCS provides support for the Space Shuttle Program into the 21st Century and a basic infrastructure upon which to base future design projects such as the Orbiter Upgrades and X-33/RLV.

The CLCS replaces the current Launch Processing System (LPS) with state-of-the-art Commercial Off the Shelf (COTS) based technology. Where task requirements can be met safely by COTS software products, the COTS software is utilized instead of developing custom applications. Any custom software that is developed is written in high level languages which have demonstrated a high degree of portability between platforms. COTS hardware is also utilized where possible in the CLCS. This strategy provides a reliable system that is modular, expandable, and extensible. It is based on open hardware and software standards, easily incorporates new technology and user developed applications, and provides inherent user interface improvements.

1.3 Related Documents

National Launch Processing System, Project Baseline (aka the "Blue Book")
Checkout and Launch Control System (CLCS) System Level Specification, April 15,1997, 84K00200

1.4 Document Overview

This document is identified as the CLCS System Design Document (SDD). The document is organized into the following sections: Introduction, Checkout and Launch Control System Architecture Overview, Real Time Processing System Architecture, Shuttle Data Center Architecture, and Simulation System Architecture. Section 1 contains the introduction to the CLCS and the SDD and includes a list of documents referenced within the SDD. The CLCS Architecture Overview section provides a broad overview of the CLCS concepts and facilities to assist in understanding the CLCS role in command and control systems applications. The remaining sections provide a more detailed discussion of the architecture of the CLCS system elements.

2. Checkout and Launch Control System Architecture Overview

The Checkout and Launch Control System (CLCS) provides the facilities for system engineers and test conductors to command, control, and monitor space vehicle systems from the start of the Shuttle Interface Test through terminal countdown and launch or Abort/Safing and Scrub Turnaround. (See Figure 1 Checkout and Launch Control System). The CLCS architecture is flexible and scaleable. CLCS has been designed to be used to used in a wide range of control and checkout applications, including future space vehicles. In addition to Space Shuttle launch processing operations, CLCS is also used for test and control functions in:

1. Cargo Integration and Test Equipment (CITE) Facility
2. Hypergolic Maintenance Facility (HMF)
3. Dryden Flight Research Center (DFRC)
4. Shuttle Avionics Integration Lab (SAIL)

This section provides an overview of the CLCS architecture, describes the CLCS hardware and software configurations, and the operational environment.

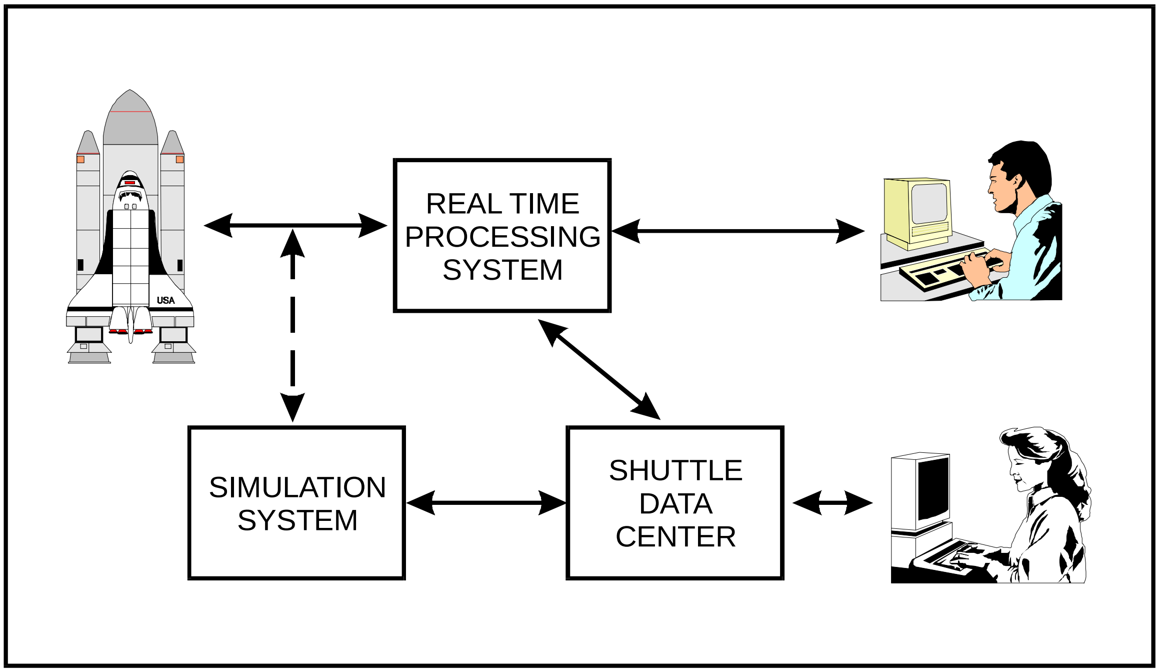


Figure 1 Checkout and Launch Control System

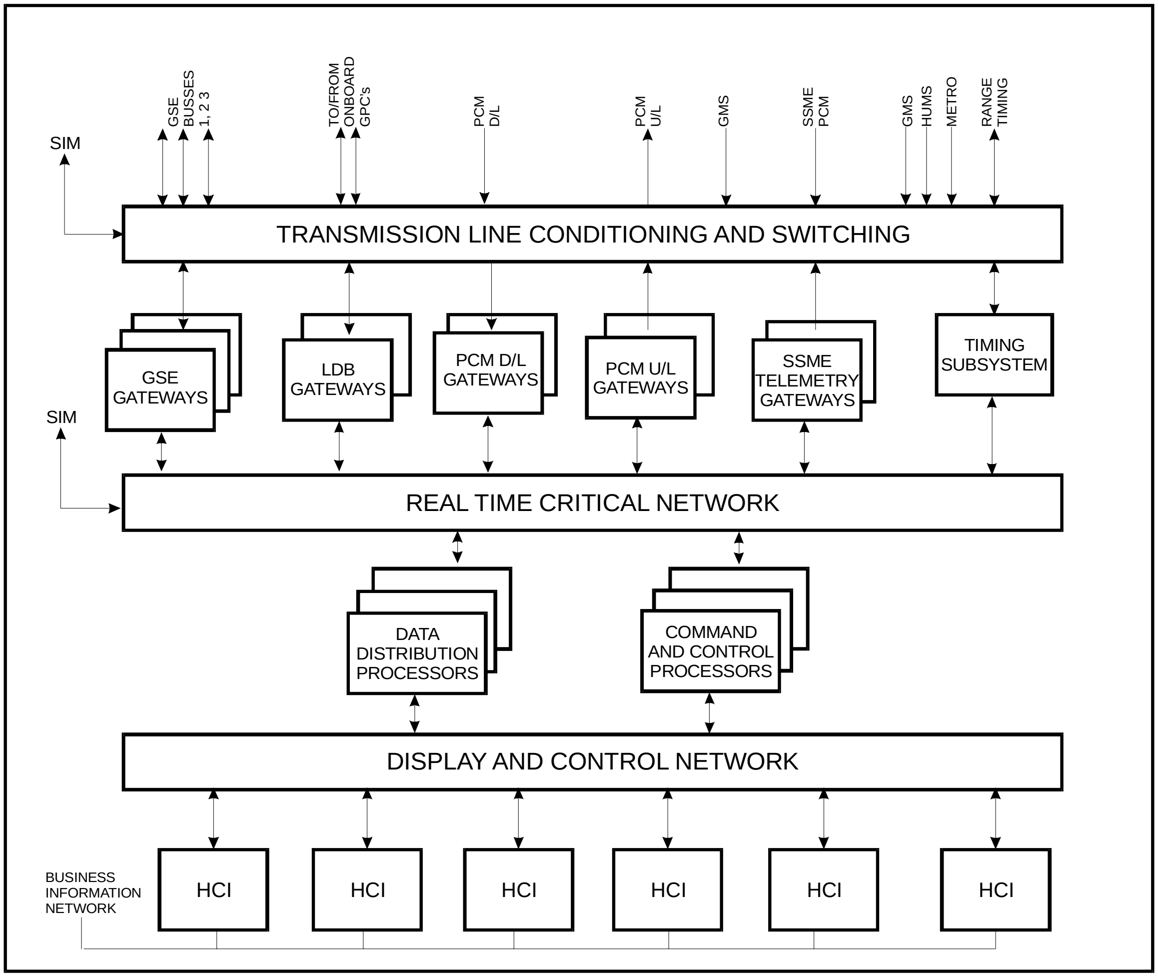


Figure 2 CLCS Architecture Overview

Figure 3 CLCS System Architecture, illustrates a top level view of the Checkout and Launch Control System. CLCS comprises three major systems:

1. Real Time Processing System
2. Shuttle Data Center
3. Simulation System

The Real Time Processing System provides the real time processing and control of flight and Ground Support Equipment (GSE) under test.

The Shuttle Data Center provides Configuration Management (CM) for system and applications software, maintains the hardware interface databases, and provides recording and playback functions during and after test operations.

The Simulation System supports the creation, modification, test, maintenance, and operation of mathematical models of the Shuttle and GSE systems which simulate the ground equipment, Orbiter functions and computers, the ET, SRBs, and payload protocols. The Simulation System provides support for the execution and interfacing of the models to the CLCS in order to provide:

1. Testing and validation of CLCS Application software
2. Checkout of CLCS equipment
3. Checkout and validation of test procedures (i.e., both manual and automated)

4. Launch team training

The simulation system may be connected directly to the RTPS interfaces and provide the equivalent hardware interface signals or may be connected to internal RTPS networks for software development purposes.

In addition to these systems, a Business Information Network provides on-line documentation access (Operation Maintenance Instructions), online paperwork generation (problem reports, engineering changes, etc.) and access to office and remote site systems.

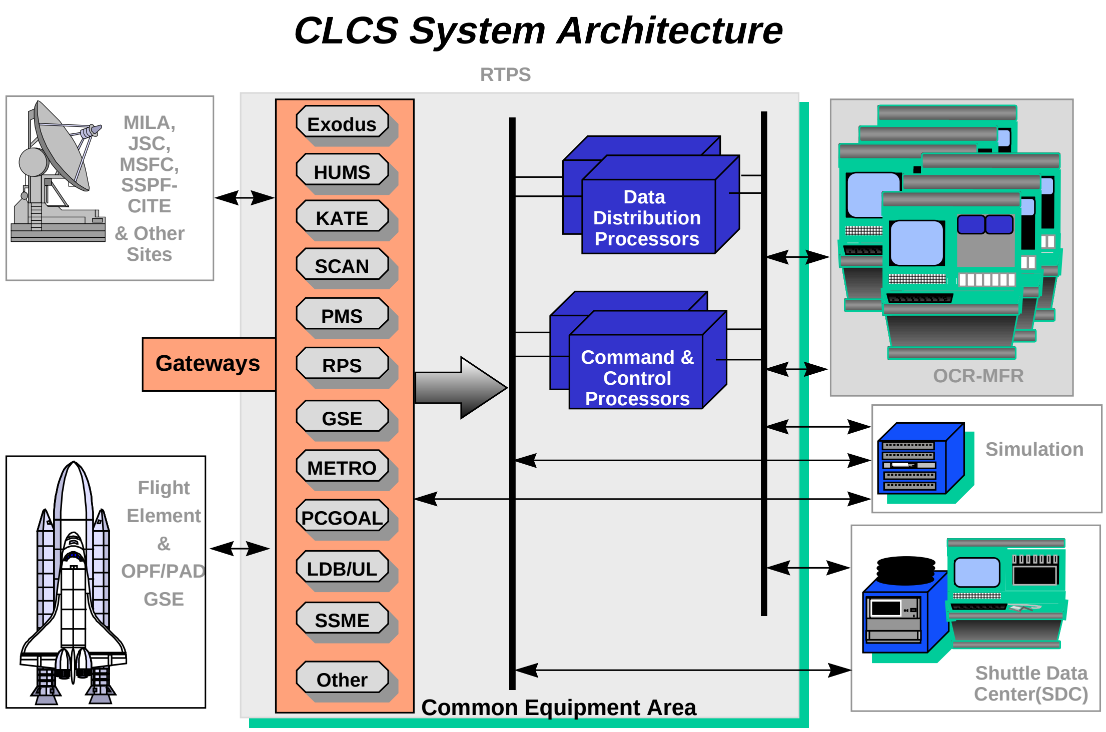


Figure 3 CLCS System Architecture

Figure 4 CLCS System and Support Software Architecture, illustrates the architecture of the software that supports RTPS test operations. The three major software groups are:

1. Support Software
2. RTPS System Software
3. User Application Software

The support software in the Shuttle Data Center prepares test applications, database information, and system software for the Real Time Processing System. A Configuration Management system provides a central repository for both system software and user application software.

The RTPS system software is the software that provides the real time, online environment and consists of Commercial Off The Shelf (COTS) operating systems, system and application services, system management services, and system unique applications. The system unique applications are those applications that provide the functional characteristics of the processing systems, e.g. Data Distribution Processing, Command and Control Processing, gateways, etc. The system applications and the application services provide the platform for the user applications to perform test and checkout functions. The system software provides the platform for user application software development and execution.

User application software commands, controls, and monitors ground and flight vehicle end item hardware. User applications consist of viewers for displaying information on consoles, test application scripts, end item managers, data fusion algorithms, and Prerequisite Control Logic.

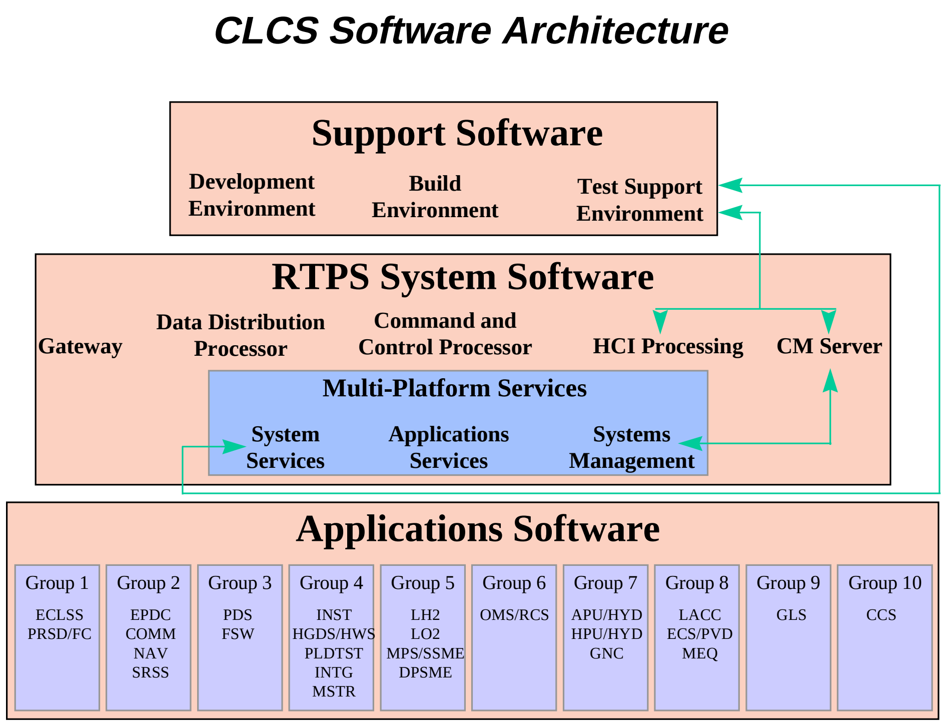


Figure 4 CLCS System and Support Software Architecture

2.1 Real Time Processing System

The Real Time Processing System (RTPS) is a closed loop control system used for test and launch operations of the Space Shuttle. The RTPS is comprised of hardware subsystems, system software and applications software. The RTPS system software and hardware provides a reliable platform for controlling the hardware interfaces to the end items and transferring commands and data between applications software and end items. The RTPS system software monitors and controls RTPS hardware subsystems, user applications monitor and control end items under test. RTPS is implemented in a fault tolerant, distributed computer system. RTPS provides the following test and control functions to applications software:

1. Command functions from the console displays and command panels
2. Information display of end item and RTPS status

3. Automated test scripts
4. Closed loop control functions for end items
5. Automated monitoring for measurement constraint violations
6. Automated capability called data fusion which combines multiple subsystem measurements to present the state and summary status of end items

These test and control functions are allocated to the functional areas in the Real Time Processing system as illustrated in Figure 5.

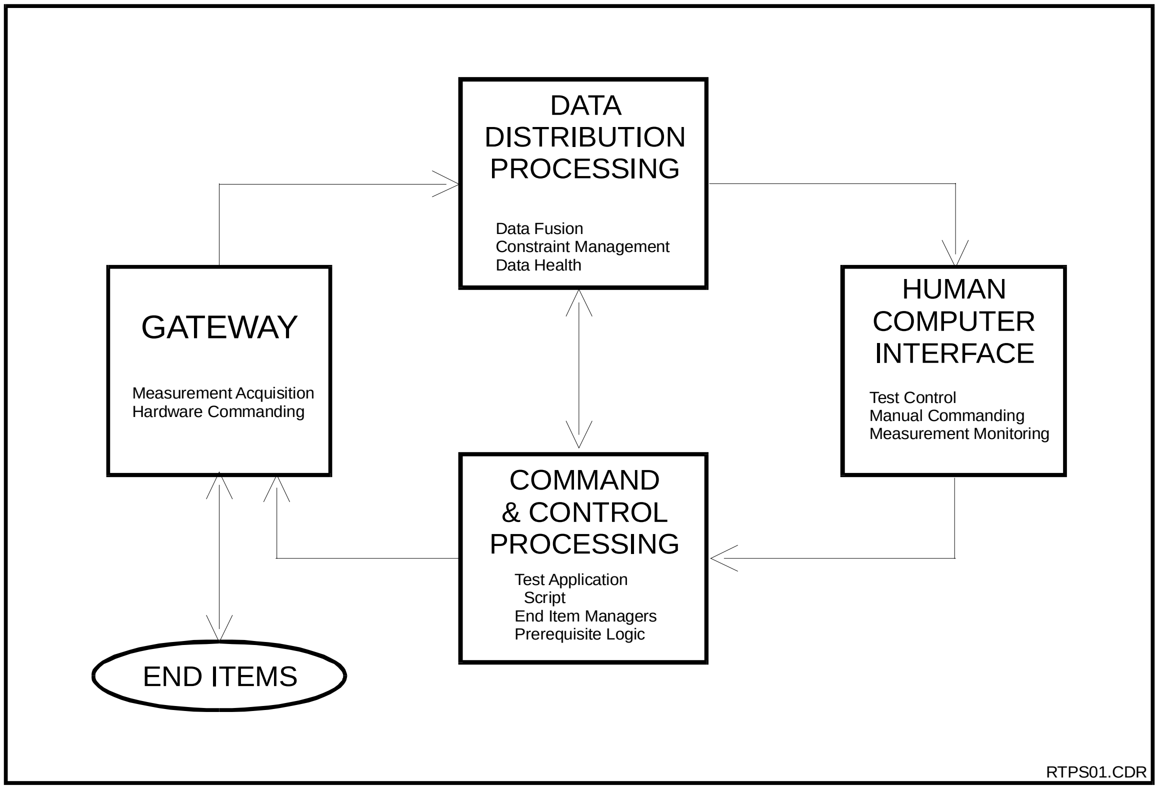


Figure 5 Real Time Processing System Functions

2.1.1 RTPS Principles

The following principles have shaped the architectural definition of RTPS. These principles provide significant operational benefits while minimizing the long term cost of operations.

Leveraged Solution: Reduce the cost of RTPS implementation by leveraging off other existing work. This includes applicable work from KSC and other NASA centers; COTS hardware, operating systems, languages and tools; and Standards ranging from ISO to ANSI to ad hoc. This represents a savings in both development and maintenance costs.

Scaleable Distributed Architecture: RTPS is based on a distributed architecture that can be scaled by increasing the capacity on a box by box level rather than having to replace the entire system. Maximum data rates across the system have been determined. Each box has been be sized to handle this maximum load in order to ensure adequate performance during peak demands.

Message Based Rather Than Storage Based: Reliable messages, rather than a common system wide data store, have been chosen as the glue that binds the system together. Reliable messaging is a well understood approach for building distributed systems and is available as a COTS solution via multiple technologies. It simplifies redundancy management within RTPS sets and increases the fidelity and quality of End Item monitoring and control.

Improved Fault Tolerance: Fault tolerance and redundancy management is extended to cover End Item user applications. These applications provide safe and effective control of the Space Shuttle and GSE. As a minimum fail-safe operation will be supported. Additional fail-operational support will be provided where practical.

Consolidated Data: RTPS consolidates data from a number of external systems, previously isolated from CCMS that provide End Item test relevant information. This will enhance the information available to control room operational personnel to aid in making informed decisions.

Reliable Data: RTPS provides delivery of complete, reliable End Item data, including data health, to users and user applications. ***Data health*** information is provided for each FD update allowing it to be tested for usability directly. Data health factors in Gateway status for the FD, knowledge of the FD's data path's health, and input from engineering. ***Reliable data delivery*** ensures that each concentrated FD update message is received resulting in no missing blocks of FD updates. FD update messages occur at a fixed periodic rate to each subscribing computer and are numbered. If a subscribing computer misses a message, retransmission can be requested. ***Complete data delivery*** ensures that an application can process all data changes for selected FD's, not just the values that existed when an application reads them. FD updates, including time of change and health, will be queued for the application and can be processed as required by the user application.

Transforming Data into Information: The individual measurement FD's from each End Item LRU provide data about the LRU but not usually any directly usable information. ***Data fusion*** combines values of multiple FD's with good data health to determine state of an LRU or other summary information to form a new FD that can be tested directly. As an example, a data fusion FD that logically combines voltage sources, switch and relay states, and bus connections can be use to represent the ON/OFF state of Orbiter power. Use of data fusion FD's greatly simplifies user application development and data retrieval.

Deterministic System Timing: RTPS provides deterministic system timing which enhances early recognition of problems, early fault detection, and eliminates latency for rapid system response.

Increased Availability: Numerous features within RTPS extend the availability and level of service. Additional control room personnel such as test directors, and remote personnel such as engineering in test/work areas can join into RTPS testing through built in access safeguards using dedicated and portable workstations.

Layered Applications: Layering allows actions to be defined and tested once and used repeatedly. As an example, data health and data fusion permit the logic of coming up with the state of an LRU to be defined once. It can be reused with confidence by many application programs, user displays, and data retrievals for years to come. This reduces the amount of application program development required and makes them more understandable.

Advanced End Item Monitoring: Any user or user application can place a ***constraint*** against any FD requesting notification should the constraint be violated. Constraint monitoring and exception notification is performed by RTPS system software at data rate speed thus allowing every sample of data to be screened. Both standard and fusion FD's can be monitored. This permits End Items to be monitored with far greater resolution and reliability. Constraint examples include: a Test Application Script requesting all OMRSD requirements be monitored and reported for each system; a Test Application Script requesting Launch Commit Criteria be monitored for GLS; and a system's End Item Managers requesting that any deviations from the current commanded system state be reported.

2.1.2 RTPS Architecture Model

The RTPS architecture is presented in this section with multiple views of the conceptual model that illustrate the functions, flexibility, and scaleability of the architecture. These views will include:

- The RTPS functions (independent of allocation to processing systems)
- RTPS as a Closed loop control system
- The Data Distribution Processing function
- The Command and Control Processing function
- The Human Computer Interface
- The Gateway function
- The network topology
- A description of a control room set
- The software and integration development (SDE's, IDE's)

2.1.2.1 The RTPS Closed Loop Control System

The CLCS Real Time Processing System provides a closed loop means for testing, monitoring, and controlling hardware end items. In RTPS, measurement data is acquired from hardware end items, time ordered, and distributed to software subsystems to perform monitor and control functions. RTPS contains 6 software subsystems which perform the closed loop control capability:

1. Data Acquisition
2. Data Fusion
3. Constraint Management
4. End Item Managers
5. Prerequisite Control Logic
6. Test Application Scripts
7. Data Viewers

The relationship between these subsystems, the hardware end item, and the test engineer are illustrated in Figure 6 RTPS Closed Loop Control.

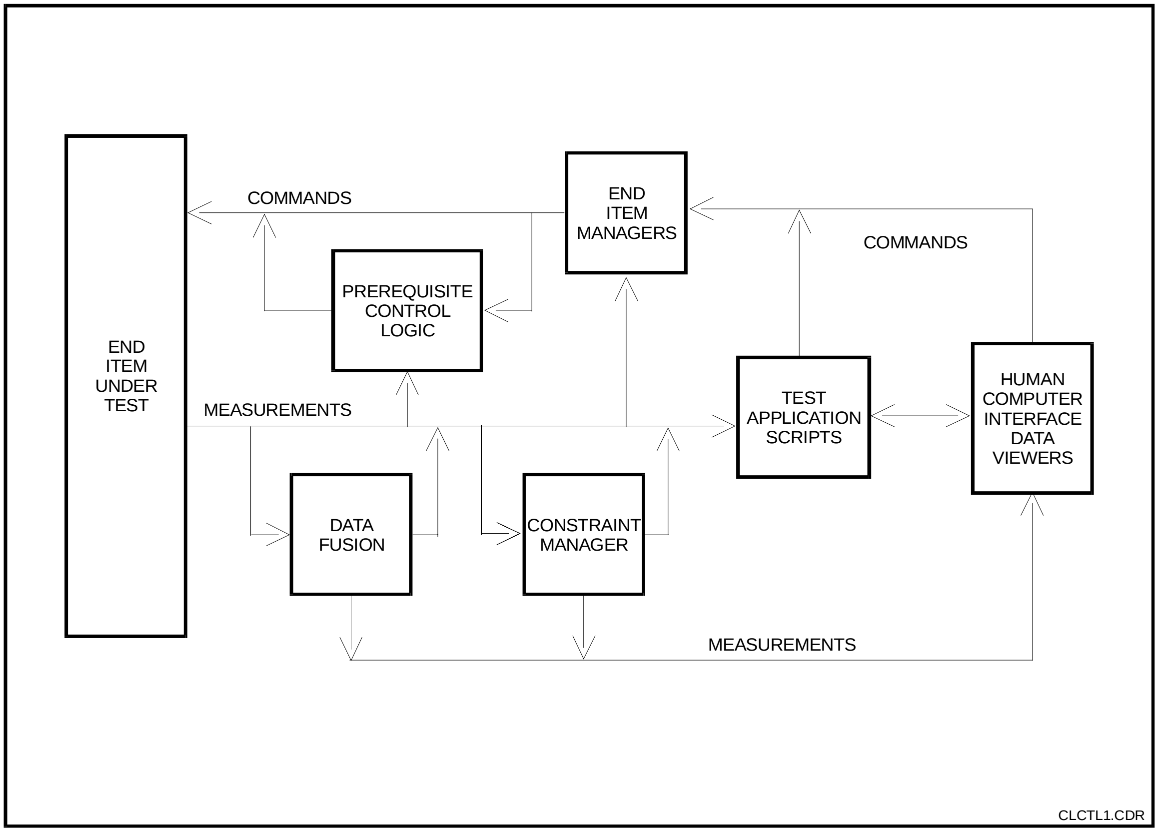


Figure 6 RTPS Closed Loop Control

Measurement data are acquired from end items, checked for change, put into proper time sequence, and distributed to the software functions for processing. System engineers, using workstation consoles in the Operations Control Room, initiate test activity from HCI positions.

The monitoring functions are Data Fusion and Constraint Management. The Data Fusion function combines multiple measurements into information which is presented to the other software functions and on user displays. Measurement health status is also used in determining the state of hardware end items. The Constraint Manager monitors measurement status and health, including fused measurement data, and notifies other application software and human users of measurement constraint violations.

The control functions are Test Application Scripts, End Item Managers, and Prerequisite Control Logic. Test Application Scripts are user defined procedures that can issue commands to test and control hardware end items. End Item Managers are user defined processes or state machines that accept command requests from Test Application Scripts or console operators. The command requests may be individual commands to the hardware end item, or may be commands that cause the end item manager to execute a sequence of hardware commands to put a subsystem into a specific state while monitoring the measurement responses to assure safe operation. For example, an End Item Manager may accept a vehicle power up command which would issue a sequence of commands to power on multiple subsystems while monitoring the power busses and status of the subsystems. Prerequisite Control Logic verifies commands to hardware end items can be issued safely.

2.1.2.2 RTPS System Level Architecture View

The RTPS software functions are implemented in a distributed, heterogeneous computer system. Figure 7 RTPS Functional Architecture, illustrates the allocation of software functions. The RTPS functional areas are:

1. Gateway Processing
2. Data Distribution Processing
3. Command and Control Processing
4. Human computer Interface

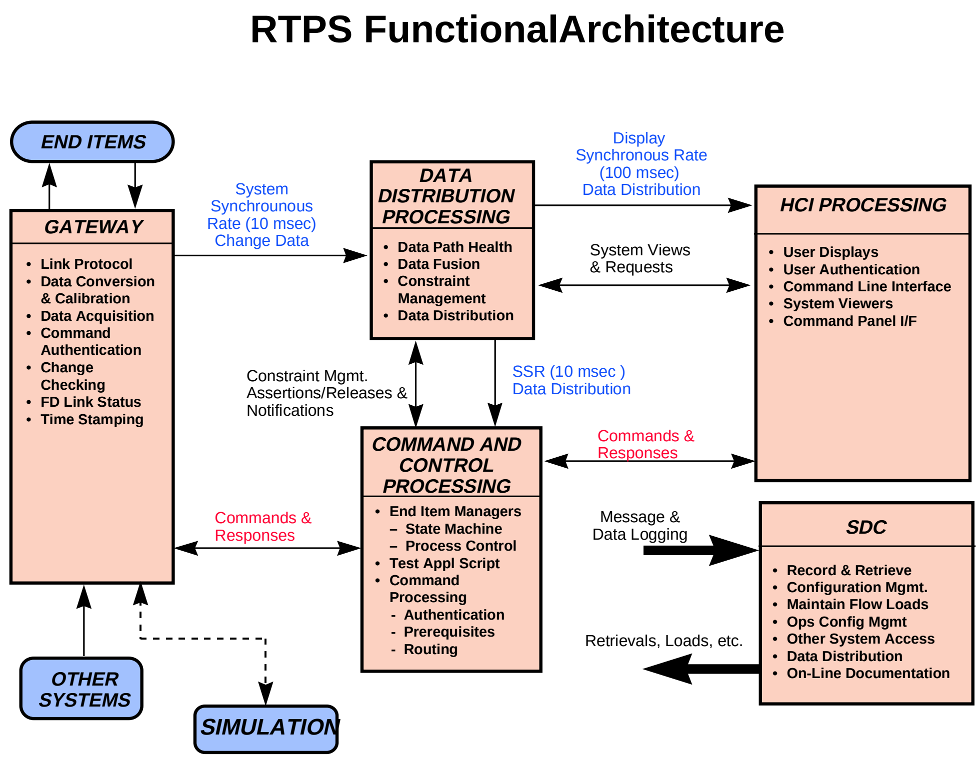


Figure 7 RTPS Functional Architecture

The functions of the Real Time Processing System software architecture described above are implemented as system applications within a framework that includes Application Services, System Services, and a standard, off the shelf operating system as illustrated in Figure 8. System Services provide a common set of services that allow the software layers above to be easily transported to other hardware/software platforms. Application Services provides access to command and measurements and other services that are unique to the control system environment. This foundation software platform provides the flexibility to change hardware platforms, and also makes possible the scaleability required to apply RTPS to a wide variety of applications. The foundation software is designed to accommodate multiple functions (DDP, CCP, HCI) in a single processing system or to allocate these functions over multiple processors in a distributed system. This architecture also has the flexibility to distribute heavy processing loads for single functions (e.g. CCP or DDP) over more than one hardware platform.

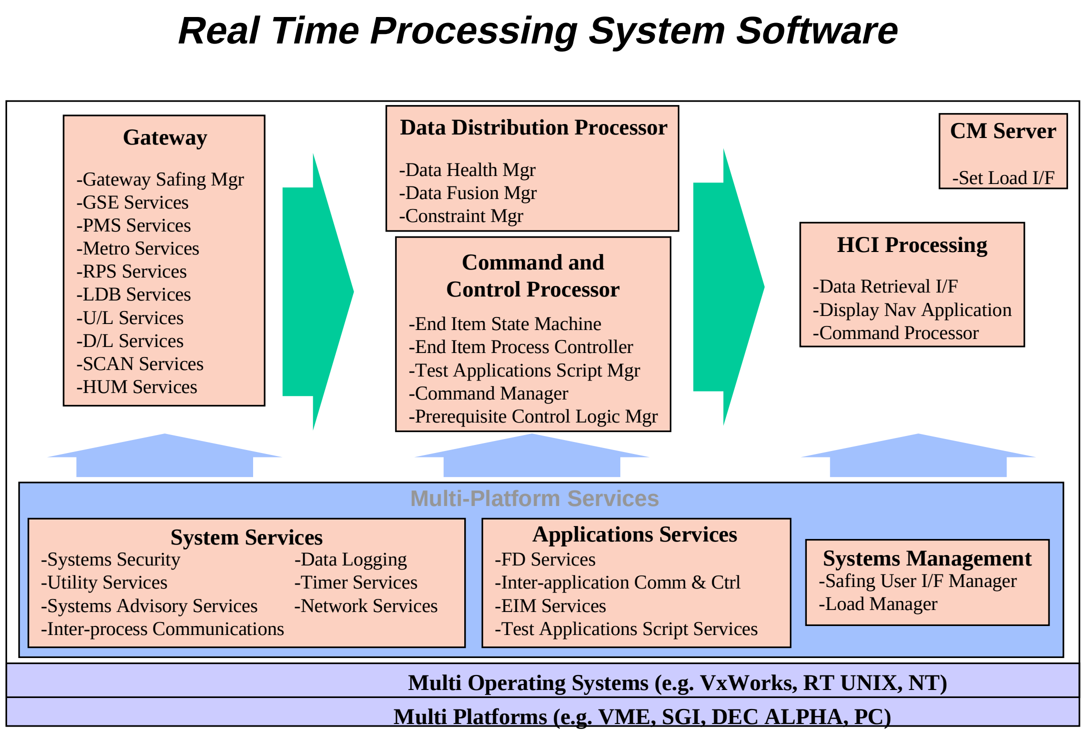


Figure 8 Real Time Processing System Software Architecture

2.1.2.3 The Gateway Function

The Gateway functions in the RTPS provide the hardware interfaces for bi-directional monitoring and control of end items. The Gateway subsystems perform the following functions:

1. Control the hardware link to the end item test article
2. Accept commands from Command and Control Function
3. Issue commands to end item
4. Acquire measurement data from the end item
5. Perform change checking on measurement data
6. Transmit measurement change data and status to Data Distribution Processing function

Figure 9 Gateway System Architecture, illustrates the generic architecture of Gateway subsystems. The Gateway System Manager manages the hardware unique interfaces to acquire measurements and issue commands. Command and Measurement Descriptor Tables (C/MDT) contain information that correlates the Function Designator Identifiers (FDID) to physical hardware addresses. Acquired measurements are checked for change against the current value (last read value). Changed measurements are calibrated, if necessary, and transmitted to the Data Distribution Processing function via the RTCN at the System Synchronous Rate (SSR). (The System Synchronous Rate is the rate at which the gateways provide updated measurement information. The SSR is a system parameter (expected to be 10 ms) which can be adjusted when RTPS is not supporting test activity.)

Commands from the Command and Control Processing Function are received via the RTCN interface. The hardware address of the command is accessed from the MDT and the command is issued to the hardware interface.

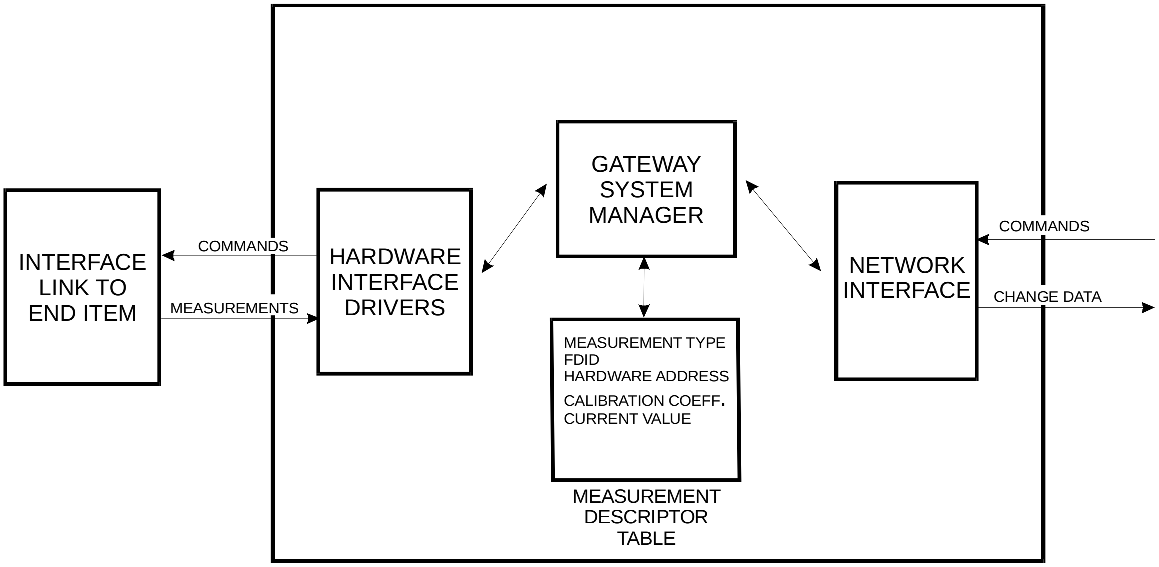


Figure 9 Gateway System Architecture

The Real Time Processing System supports Test Sets in multiple physical locations for processing the Space Shuttle. The hardware interfaces at each of these facilities vary in number and type. The RTPS Gateway systems provides the following interfaces to end items under test:

1. Ground Support Equipment (GSE)
2. Orbiter Launch Data Bus (LDB)
3. Orbiter PCM Downlink
4. Orbiter PCM Uplink
5. Shuttle Main Engine PCM Downlink
6. Ground Measurement System (GMS)
7. Pad Meteorological Data (Metro)
8. Hydrogen Umbilical Mass Spectrometer (HUMS)
9. Shuttle Data Stream (SDS)

2.1.2.4 Data Distribution Processing Function

Data Distribution Processing performs the following functions:

1. Accepts input measurement change data from Gateways
2. Merges all the Gateway change data steams into one time ordered stream of measurement changes
3. Appends health and status information to measurement data
4. Distributes changes to:
   a) Constraint Function at the SSR
   b) Data Fusion Function at the SSR
   c) Data Health Function at the SSR
   d) Human Computer Interface function at the DSR
   e) System and user test applications at the SSR

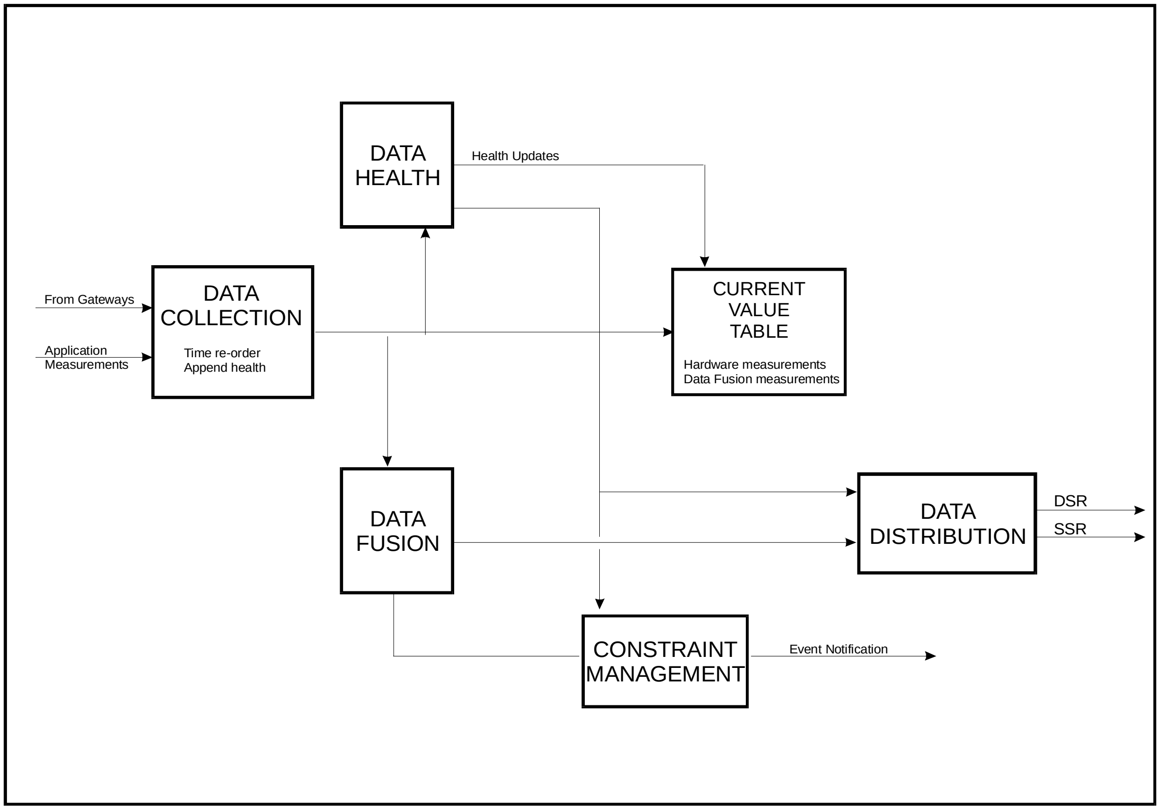


Figure 10 Data Processing Architecture

Figure 10 illustrates the Data Processing function architecture. The Data Collection component receives hardware measurement changes from gateways and application software measurement changes via the network interface or from local applications. The Current Value Table (CVT) is updated with the new measurement value. The Data Health component updates measurement health status in the CVT whenever a change in health is received. Data Fusion uses input change data to recalculate the value of Fused Function Designators which are then distributed in the same manner as hardware measurements. The Constraint Manager, on request, monitors both hardware and application derived measurements for values that violate user defined constraints. Violations are reported as event notifications. The Data Distribution component distributes measurement changes to the Command and Control Processing function and to the Human Computer Interface (HCI).

The Data Distribution function also provides the CVT maintenance services within the CCP and HCI.

2.1.2.5 Command and Control Processing Function

The Command and Control Processing performs the following functions:

1. Accepts measurement data and health changes from Data Distribution
2. Provides measurement data and health changes to user test applications and system applications for processing
3. Provides the services and environment for user test application scripts to execute
4. Executes End Item Managers
5. Accepts event notification from the Constraint Function and provides the notification to the appropriate application
6. Provides user applications the capability to set constraints
7. Accepts HCI and application commands
8. Authenticates and routes commands to the appropriate application or Gateway function

9. Performs Prerequisite Control Logic sequence if required

Figure 11 illustrates the Command and Control Processing architecture. Commands are received from the user's console to execute Test Application Scripts or send commands to the end item test article. Test Scripts provide the capability to automatically execute sequences of commands to perform test functions. Test Application Scripts may perform conditional tests on measurement parameters to alter the test flow. Multiple Test Application Scripts may run concurrently. End Item Managers accept commands to perform high level functions, e.g. bring up the electrical power distribution system. There are End Item Managers for all major vehicle and ground support systems. End Item Managers are object oriented, state or process bases control systems. A high level End Item Manager may have one or more subordinate End Item Managers that control individual components of the system under control. Unlike Test Application Scripts, End Item Managers issue commands to the hardware end item. Some commands to the hardware end item may have Prerequisite Control Logic sequences that perform safety checks before issuing the command to the hardware end item. Prerequisite sequences perform logical checks on measurement parameters to determine if the intended command can be safely issued. The source for every command is authenticated before issuing the command.

Measurement change data sent by Data Distribution Processing is input to the CCP functions; End Item Managers, Test Application Scripts, and Prerequisite Control Logic.

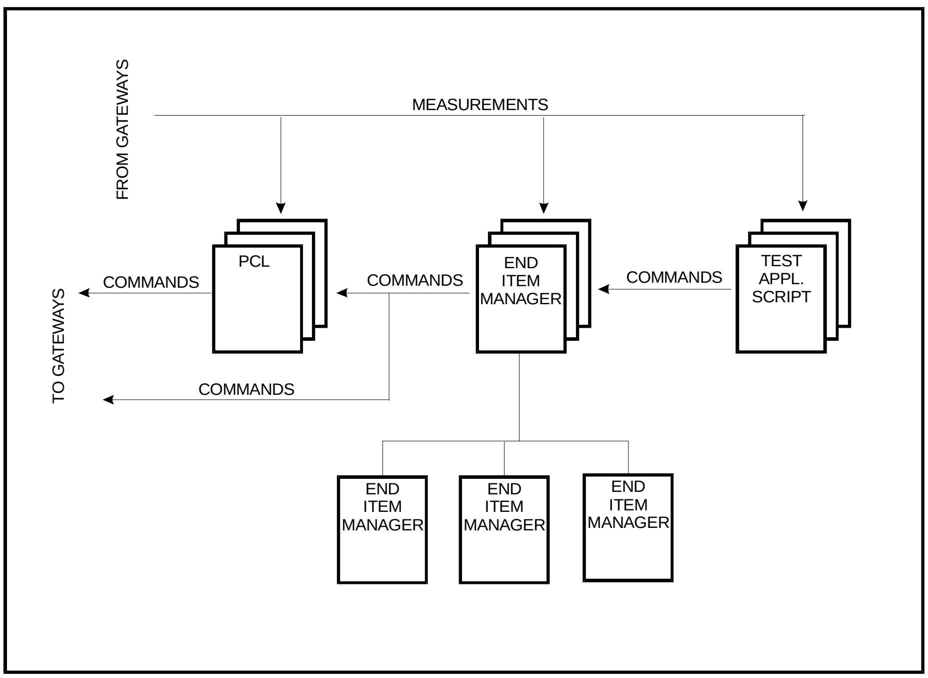


Figure 11 Command and Control Processing Architecture

2.1.2.6 The Human Computer Interface Function

The Real Time Processing System console (Human Computer Interface) architecture is illustrated in Figure 12. The Human Computer Interface performs the following functions:

1. Provides display information to user
2. Accepts user input from:
   a) Keyboard
   b) Pointing device
   c) Command Panel
3. Acquires data for display by:

a) System viewers
b) Dynamic Data Visualization tool

In addition to the real time control function, the console has the provisions to access the Shuttle Data Center software and documentation repositories, data archives, and software applications. Access to the Internet is provided via a security firewall. Operational Television (OTV) may be displayed on two of the console monitors. Controls are provided to allow users to select, pan, and zoom remote cameras. The real time monitor positions allow users to view data from other test activities that are in progress.

Each console position has a command panel which provides a safing interface for safety critical end item management.

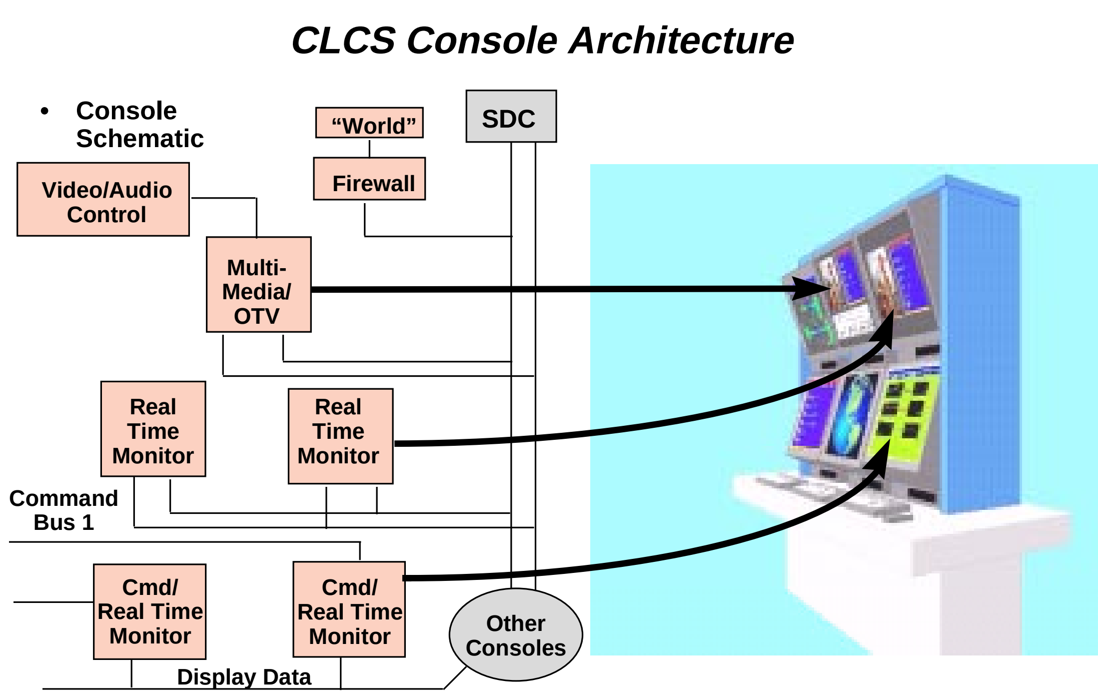


Figure 12 RTPS Console Architecture

NOTE NEED A PICTURE OF A TYPICAL DISPLAY TO DESCRIBE THE FUNCTIONS THE USER HAS

2.1.2.7 Application Software Architecture

The CLCS application software hierarchy is illustrated in Figure 13. This layered approach to application software provides the user with a reliable means of monitoring end item measurement information and provides positive control over the end item under test. The application software layers include the following user defined applications:

1. User Interface for Monitoring and Commanding
2. Test Application Scripts
3. End Item Managers

4. Constraint Manager
5. Prerequisite Control Logic
6. Data Fusion

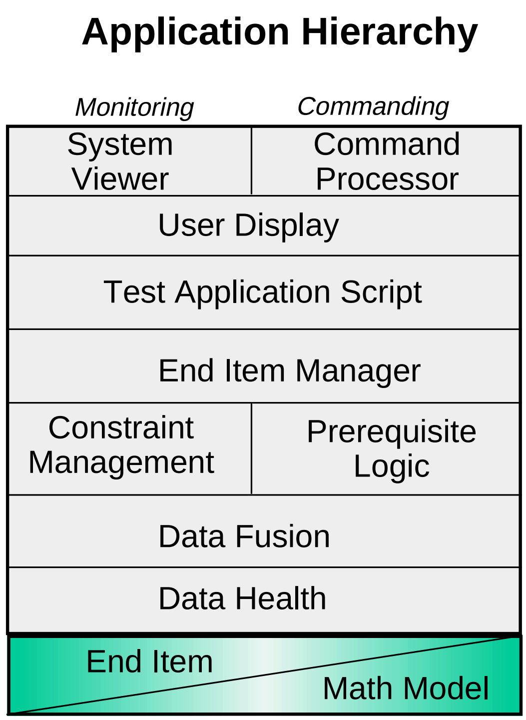


Figure 13 CLCS Application Software Hierarchy

2.1.2.7.1 User Interface for Monitoring and Commanding

The user interface is provided in the form of Viewers or display applications which are used for issuing commands, monitoring test end item measurements, application software, and CLCS system configuration and status. The following viewers provide a standardized means to display information related to end items, applications and CLCS system information:

1. Function Designator Viewer
2. End Item Manager Viewer
3. Constraint Management Viewer
4. Test Application Script Viewer
5. Data Fusion Viewer
6. Data Health Viewer
7. System Configuration Viewer

In addition to display viewers, the user also has access to the command processor via command line interface. Individual commands may be entered via the keyboard and executed.

2.1.2.7.2 Test Application Scripts

Test Application Scripts are control procedures that issue sequences of commands to end items to perform tests and to configure vehicle systems for launch. Test Application Scripts issue command requests to End Item Managers, requests to assert/release constraints to the Constraint Manager, prompts the console user for manual steps to be performed, and request execution of other Test Application Scripts. Test Application Scripts are executed as part of the Command and Control Processing function under the control of the Test Application Script Manager. The user initiates and controls Test Application Scripts via the Test Application Script Viewer.

2.1.2.7.3 End Item Managers

End Item Managers are object oriented, state based or process control applications which control and monitor test end items. End Item Managers for a hardware end item may have a hierarchical structure, where one End Item Manager controls the entire hardware subsystem with lower level End Item Managers controlling individual components of the hardware subsystem. For example, in a liquid oxygen or hydrogen system, one End Item Manager may be in control of the entire LOX or LH2 subsystem, with separate End Item Managers controlling individual valves and pumps. End Item Managers receive requests from user displays, Test Application Scripts, or other End Item Managers. End Item Managers also receive end item measurement information from Data Distribution and notification of constraint violations from the Constraint Manager. Critical control functions are implemented in End Item Managers that have high (pre-emptive) priority. These high priority control functions were called Reactive Control Logic in previous systems (LPS CCMS). End Item Managers are executed as part of the Command and Control Processing function.

2.1.2.7.4 Constraint Manager

The Constraint Manager provides a measurement monitoring service for the user viewers, End Item Managers, Data Fusion, and Test Application Scripts. On request, the Constraint Manager begins monitoring hardware subsystem measurement information (Function Designators), and software derived measurement information (also Function Designators). The requester identifies the constraints to be placed on the measurement and registers with the Constraint Manager to obtain notification whenever the measurement violates the constraint. The constraint limits may be placed on the measurement data itself, or the health of the measurement data.

2.1.2.7.5 Prerequisite Control Logic

Prerequisite Control Logic sequences provide the user the capability to define a set of conditions that must be met prior to issuing a command to the hardware end item. All end item hardware commands issued by console users, End Item Managers, or Test Application Scripts are checked for Prerequisite Control Logic sequences. If a prerequisite sequence exists for that command, the command is forwarded to the prerequisite sequence. The prerequisite sequence verifies that the command can be issued safely as defined by the user. If not, the command is rejected and the user is notified. The user may choose to override the prerequisite sequence and issue the command.

2.1.2.7.6 Data Fusion

Data Fusion is the process of combining measurement data, measurement health, and RTPS system parameters into information that may be used in the same manner as measurement data. Data Fusion outputs are known as Data Fusion function designators. User applications and system viewers may register for notification of change to a Data Fusion Function Designator. Data Fusion allows users to specify logical and algebraic equations as functions of hardware measurements, measurement health, other derived measurements and RTPS system parameters. The resulting Data Fusion function designator can then represent a higher level, or summary status of the state of a hardware subsystem. For example, a Data Fusion function designator could be defined for APU Ready for Start. When a small set of measurements contain the proper values, the APU Ready for Start derived measurement would be true, otherwise it is false. When used in this manner, Data Fusion provides the capability to define a subsystem state in one and only one place, making that information available to multiple application users. This eliminates the need for applications to redundantly check the same set of information, reduces the opportunity for errors, and minimizes the impact of changes.

2.1.2.7.7 Re-Engineering LPS CCMS GOAL Applications

The Launch Processing System CCMS subsystem has been used since the beginning of the Space Shuttle Program for test and launch operations. There is a large base of application software, written in the Ground Operations Aerospace Language (GOAL) which must be replaced with new applications that provide the same end results; vehicle checkout and launch. Advances in both hardware and software technology make it possible to re-engineer these applications in a more efficient and cost effective manner. Figure 14 illustrates the mapping of functions presently implemented in GOAL applications to the CLCS environment.

In the process of re-engineering the GOAL applications, the Data Fusion and End Item Manager functions will be used to eliminate as many redundant blocks of control and monitor GOAL code as possible. Wherever the summary status of a vehicle subsystem is needed that requires reading multiple measurements, a Data Fusion Function Designator will be used to calculate that value once, allowing it to be used by multiple applications, including Constraint Management. In a similar manner, redundant code to configure or control a vehicle subsystem will be implemented in a single End Item Manager and reuse by multiple applications.

Exception monitoring functions that are implemented in GOAL applications will be assigned to the Constraint Manager function. The resulting constraint violations are routed to End Item Managers, Test Application Scripts, and Data Fusion. CCMS Reactive Control Sequences are implemented in End Item Managers.

Display function that are implemented in GOAL applications will be implemented using the CLCS display viewer functions. These display viewers will simplify the process of creating display screen images and menus need for monitoring and controlling end items. The display viewers will also simplify the standardization of format, color, and content of displays.

Test sequences that were implemented in GOAL will be developed using Test Application Scripts. The Test Application Scripts will be significantly smaller and simpler because of the level of information that must be managed. Test Application Scripts will rely on End Item Managers to activate, configure, and control end items. Data Fusion Function Designators will provide summary status of end items, reducing the number of measurements that must be tested and the number of calculations that must be performed.

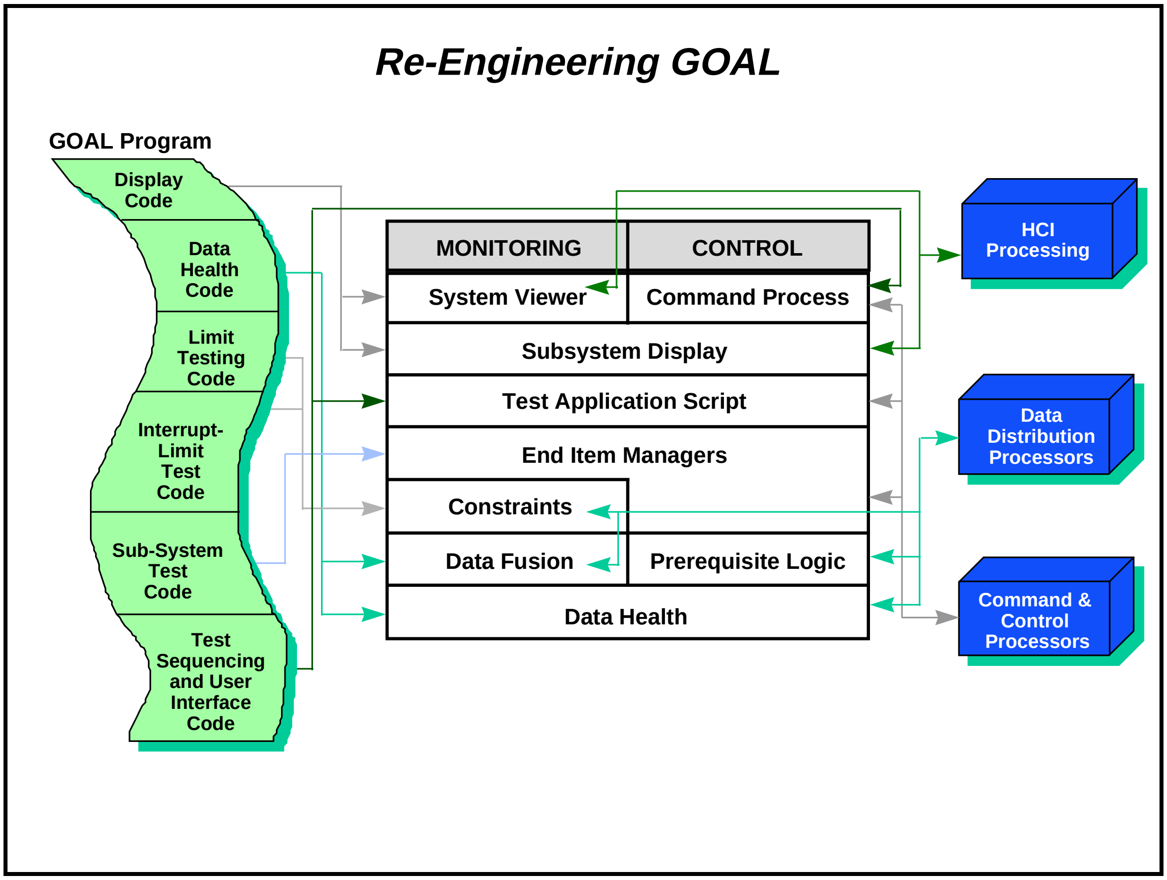


Figure 14 Re-Engineering GOAL Applications

2.1.3 RTPS Data Flows

The top level data flow for the RTPS is illustrated in Figure 15 on page 27. The following sequence describes the flow of measurement and command information in RTPS hardware:

1. Measurement Data is acquired from hardware End Items by Gateways
   a) Gateways time tag measurement data and convert to counts and engineering units
2. Gateways distribute data to DDP at the System Synchronous Rate (10 ms baseline)
   a) DDP fuses data
   b) checks constraints
   c) Adds health
   d) Consolidates and time reorders measurement data
3. DDP distributes data to CCP
4. DDP distributes data to HCI
5. SDC records traffic on the RTCN
6. HCI sends End Item Commands to CCP
7. CCP sends End Item Commands to Gateway
8. Gateway issues End Item Commands to hardware End Items
9. SDC records traffic on the DCN
10. CCP sends event notification to HCI

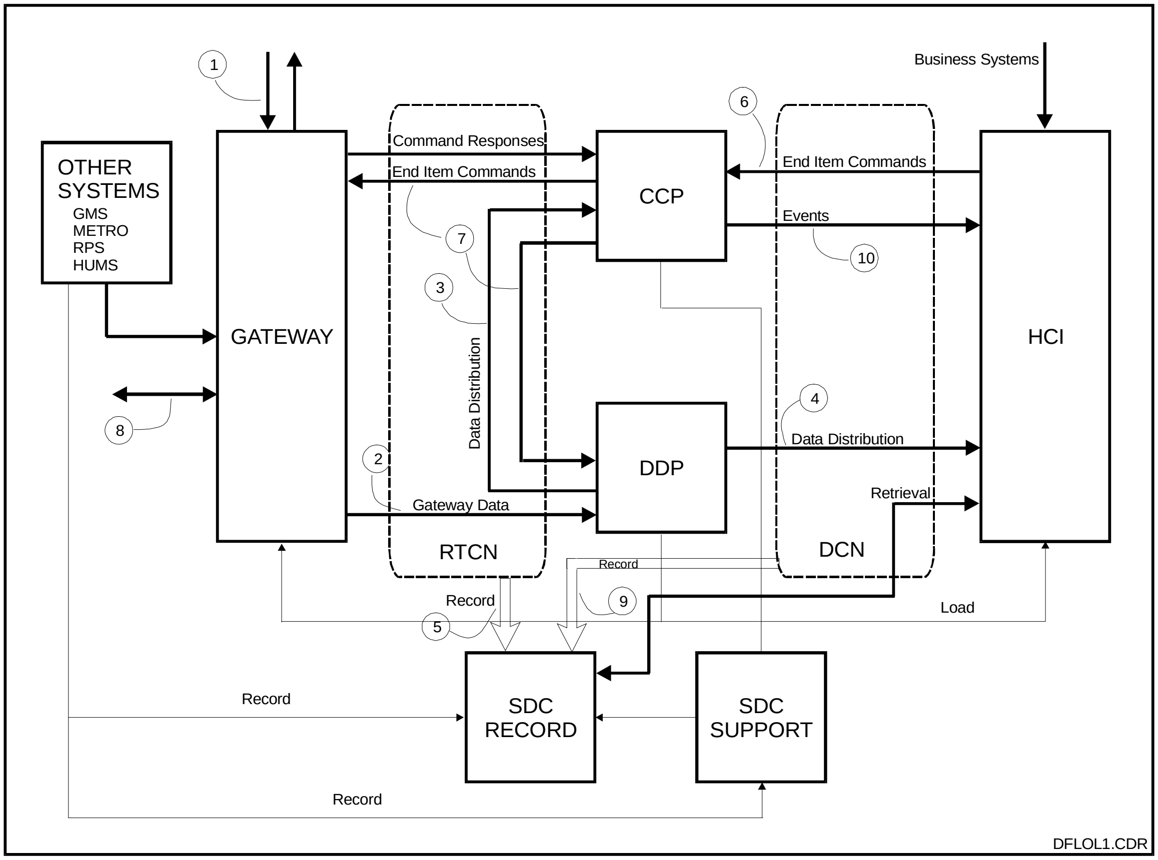


Figure 15 RTPS Data Flow

The logical flow of commands and data through the RTPS software functions is illustrated in Figure 16. Commands may originate from the user console, test scripts or End Item Managers. Commands to hardware end items may require prerequisite checks by Prerequisite Control Logic (PCL) before being sent to the gateway subsystem. The gateway subsystem issues the command to the end item or forwards it up the LDB or PCM Uplink to the Orbiter computers. Measurement data is acquired from the hardware end item and forwarded to the Data Distribution Processing function. After the measurement data is time reordered, the Data Fusion function processes selected measurements and creates new, fused data function designators. These function designators are then processed in the same manner as the hardware measurement information. The Data Health function updates the health status of hardware measurement data based on information regarding the hardware interface link, the Gateway that acquired the data, the wiring connections in the hardware, and the status of the hardware subsystem. Constraint management monitors measurement data and health status and alerts application software and users of measurement parameters with constraint violations.

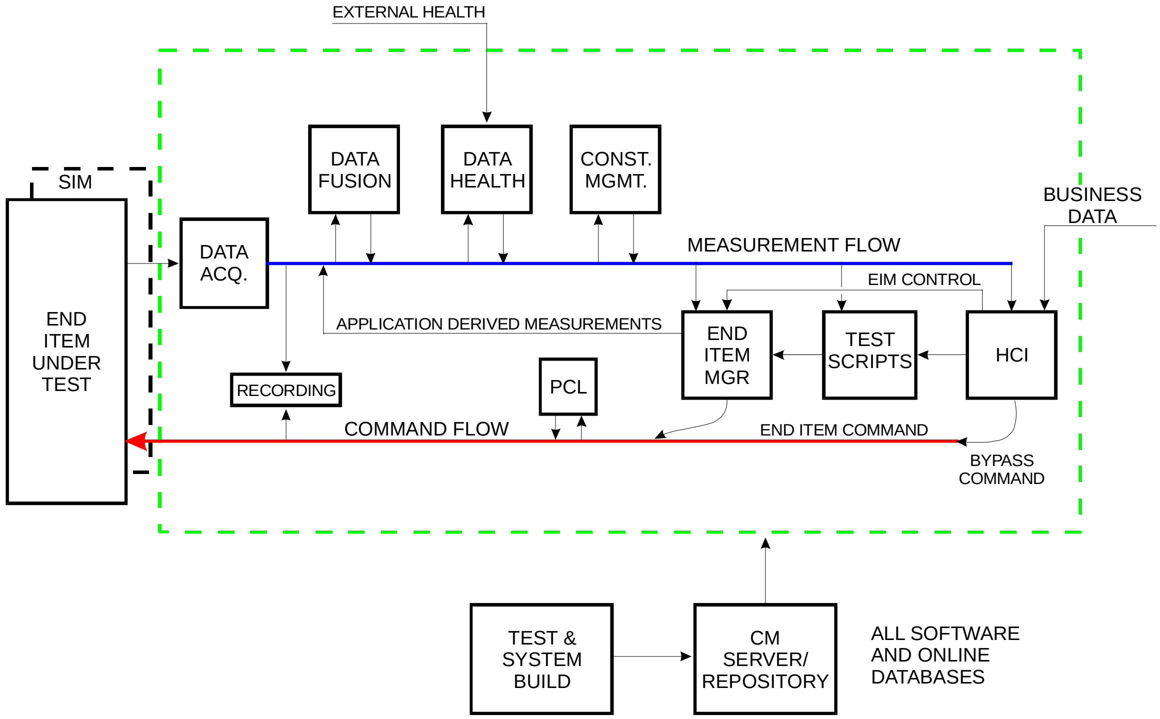


Figure 16 RTPS Data Flow

2.1.4 RTPS Physical Implementation

The physical implementations that embody the RTPS architecture vary between physical locations and type of application. In the largest, most complex, and most demanding application, Space Shuttle launch operations, a large number RTPS processing systems are available in one geographic area. In smaller, less demanding environments, a small number of processing systems are used to accomplish less complex, less critical tasks. To accommodate the varying demands, the RTPS architecture allows flexibility in grouping processing systems and processing functions to allow small to large scale installations without the need to redesign and rebuild the software systems.

In the Front End Zone, Gateway systems are arranged in groups. The groups are defined to meet the needs of specific physical test environments based on the hardware interfaces in that environment. A gateway group may include a set back-ups and spares for a specific test facilities.

Control groups consist of the combination of processing systems (DDP and CCP) and system software required to implement the Data Distribution Processing function and the Command and Control Processing function. In some environments, a single computer system may contain both the DDP and CCP functions. In other environments, e. g. the launch control center, multiple groups of CCP and DDP processors are required.

In the Flow Zone, consoles that provide the Human Computer Interface, are typically implemented in separate, individual workstations. The workstations all contain the same system software; application software and tables are loaded by engineering discipline.

A collection of these groups, interconnected by high speed local area networks, make up a Test Set. A Test Set is, therefore, all of the processing equipment, loaded with the appropriate system and application software to perform a test activity. Each Test Set contains a dedicated function, called the Test Set Master function, that is the focal point for control of all activity in that Test Set.

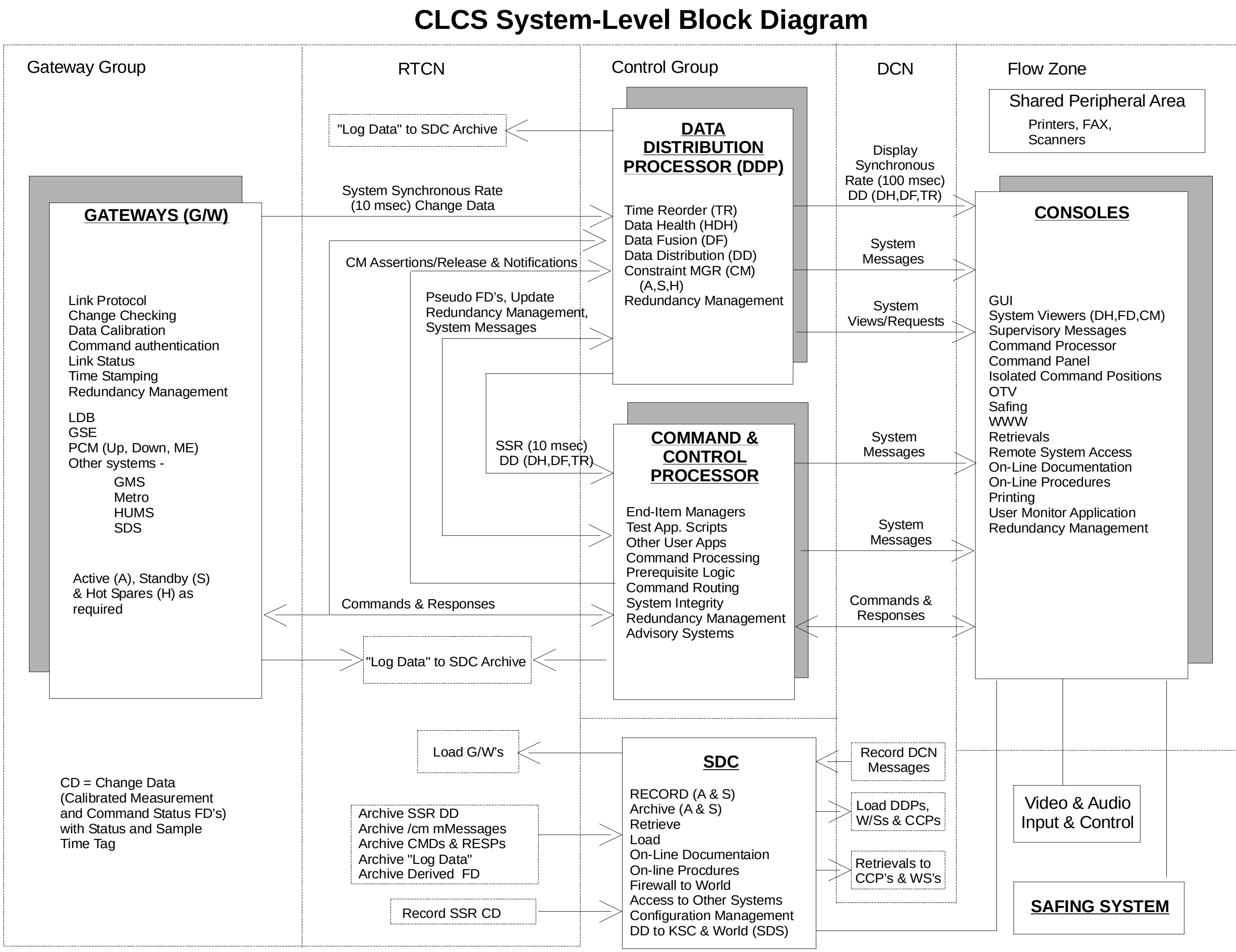


Figure 17 RTPS Test Set Physical Implementation

Figure 17 illustrates a RTPS Test Set physical implementation with a gateway group, a control group and console interfaces. In this implementation, the DDP and CCP functions are resident in separate processing systems. The shaded copies of each of the processing systems indicates the capability of providing redundant functions or spreading the processing load across multiple hardware systems. In a critical application, the active gateway and control processing systems may be replicated in a standby and "hot" spare as indicated by the "A, S, H" designation.

Multiple hardware configurations have been defined to meet the requirements for different processing environments. The hardware configurations are implemented using variations of the following configurations:

- RTPS standard configuration
- RTPS limited configuration
- RTPS minimal configuration

In addition to these configurations, development and integration configurations have been defined to support system and applications debug and integration. These configurations are:

- Satellite Development Environment
- Integration Development Environment

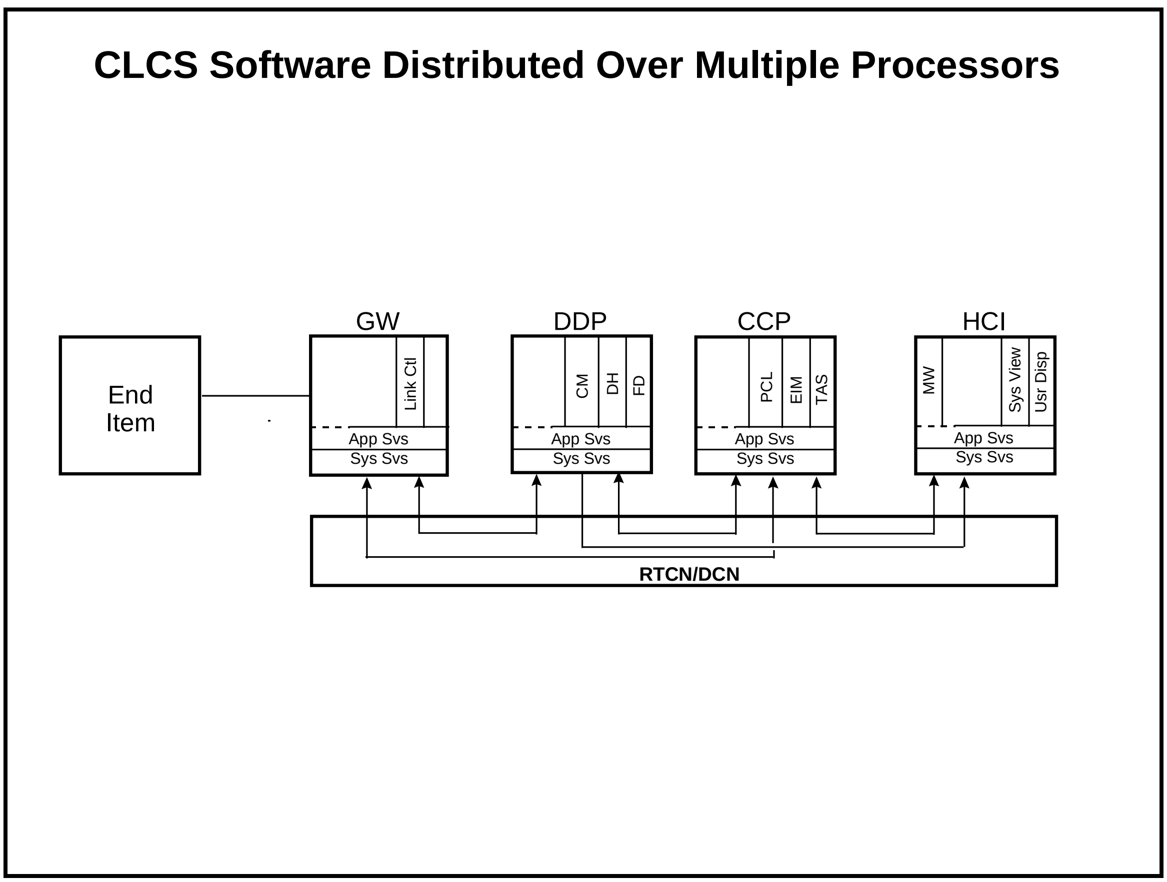


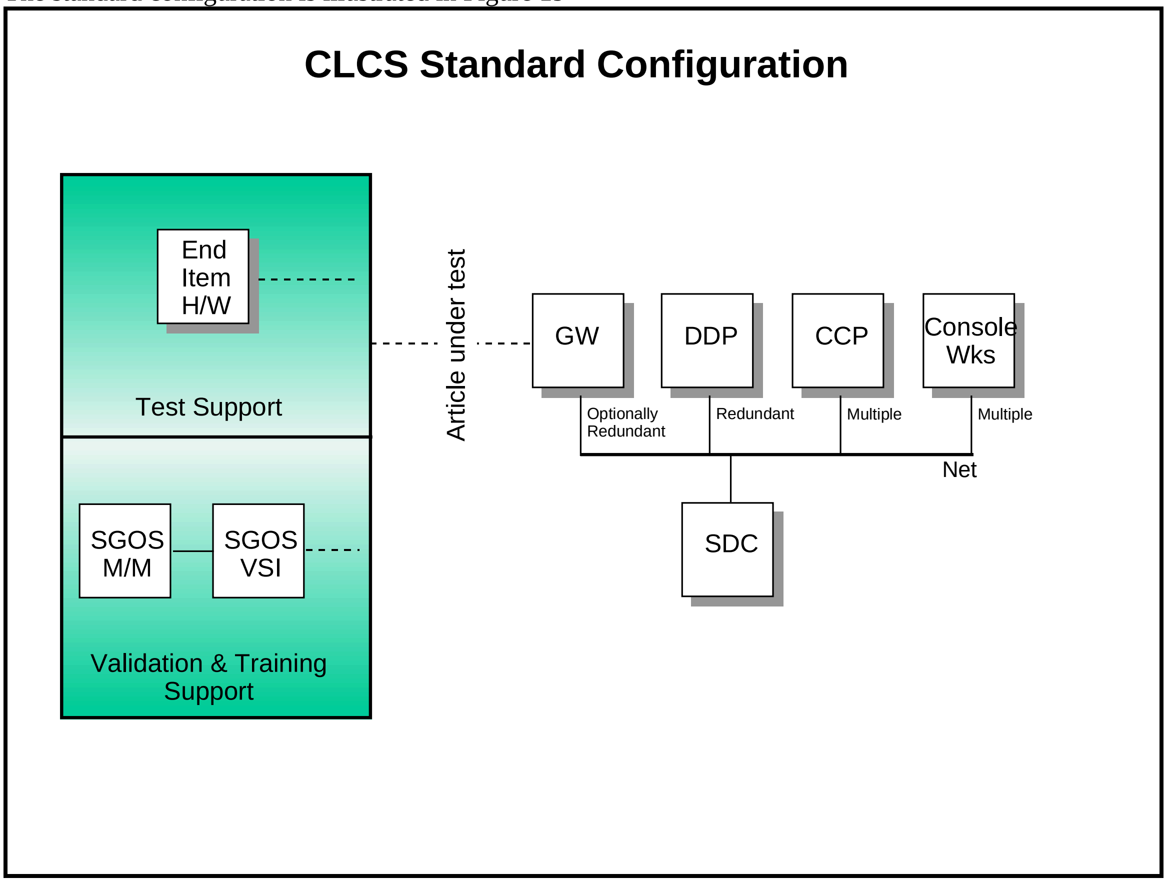
Figure 18 RTPS Software Distributed Over Multiple Processors

2.1.4.1 RTPS Standard Configuration

The RTPS standard configuration consists of single, multiple, or redundant processing systems allocated to each of the processing functions:

- Gateway
- Data Distribution Processing
- Command and Control Processing
- Human Computer Interface
- SDC Operations Configuration Manager

The standard configuration is illustrated in Figure 19



.

Figure 19 RTPS Standard Configuration

2.1.4.2 RTPS Limited Configuration

A CLCS set may be implemented in limited configuration as illustrated in Figure 20. In this environment the Data Distribution Processing and Command and Control reside in the same hardware subsystem. The limited configuration is used for application and system software development, and allows multiple users to integrate software packages. The gateway systems can be connected to the Simulation System Video Simulation Interface (VSI) or the gateways can be bypassed and simulation data injected directly on the RTCN.

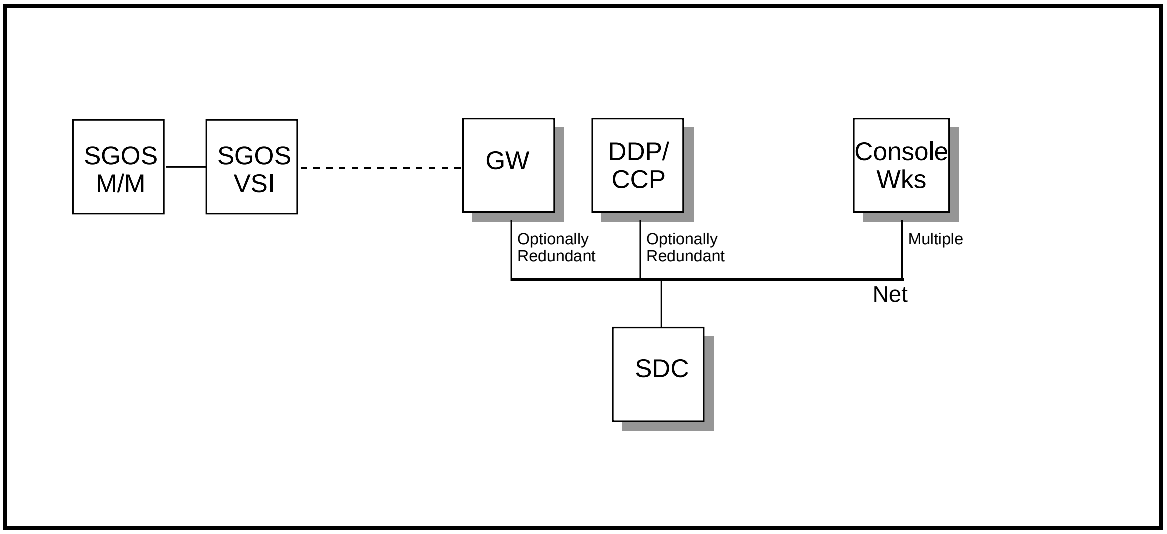


Figure 20 RTPS Limited Configuration

2.1.4.3 RTPS Minimal Configuration

The RTPS may also be implemented in a minimal configuration as illustrated in Figure 21. In the minimal configuration, the Human Computer Interface function is implemented in the same processing system as the DDP and CCP functions. In this environment, the console is a single user display/keyboard??? This environment is efficient for system and application software debug. A minimal configuration system is illustrated in Figure 21. In this configuration, users may operate locally via an attached terminal or at remote workstations attached via a network.

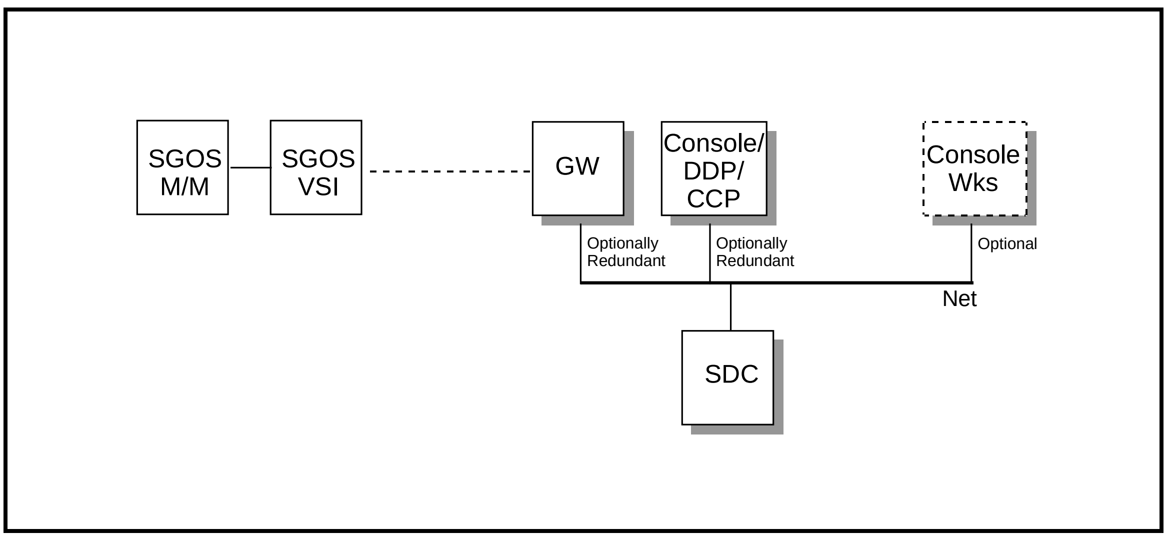


Figure 21 RTPS Minimal Configuration

2.1.4.4 RTPS Integration and Development Environment

The software development and integration environments are implemented with variations of the standard hardware configuration. These environments are connected to the simulation system interface instead of physical hardware interfaces for test and verification purposes.

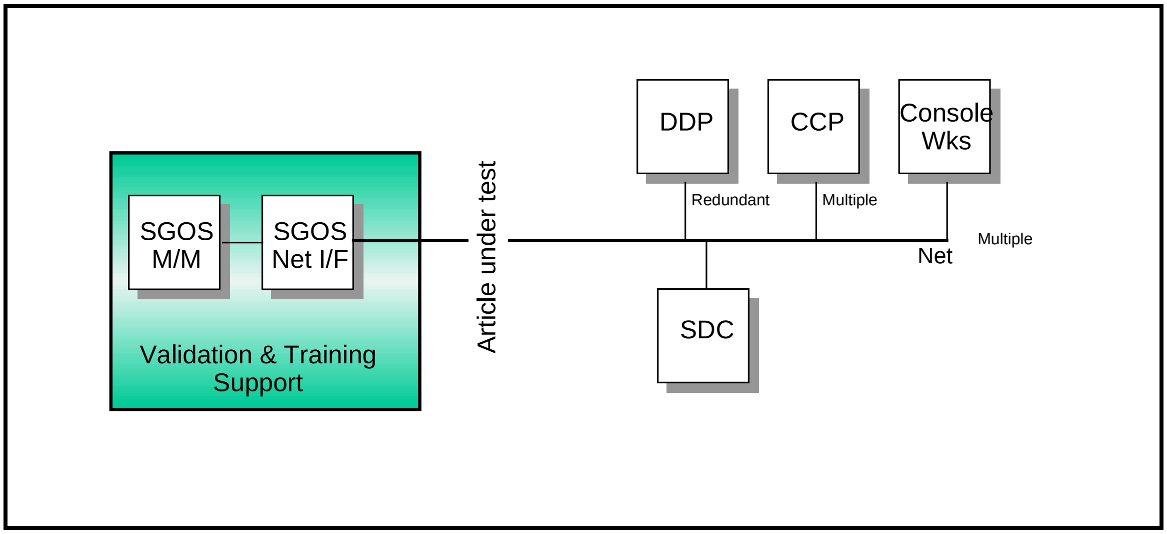


Figure 22 CLCS Integration and Debug Environment

The Launch Control Center RTPS Set is located on the third floor of the Launch Control Center. This area is divided into a Common Equipment Area (CEA) and three operations control rooms designated OCR1, OCR2, and MFR (Multi-Function Room). The LCC Set is illustrated in Figure 23. The Operations Control Rooms contain the console workstations (HCI) and shared I/O equipment, including printers, plotters, fax machines, etc. The Common Equipment Area contains control group and gateway processing systems, network switching equipment and hardware link interface patching and switching equipment.

The network patching and switching equipment allows the LCC Set to configured as multiple test sets as illustrated in Figure 24. The Set Master in the Common Equipment Area is used by operations personnel to configure and monitor the Test Sets. Groups of consoles can be attached via the DCN to Control Groups in the CEA. The Control Groups are attached via the RTCN to a group of gateways, which in turn, support specific vehicle and Ground Support Equipment interfaces. Test Sets can be physically isolated from each other to avoid network overloading and data contamination between Test Sets.

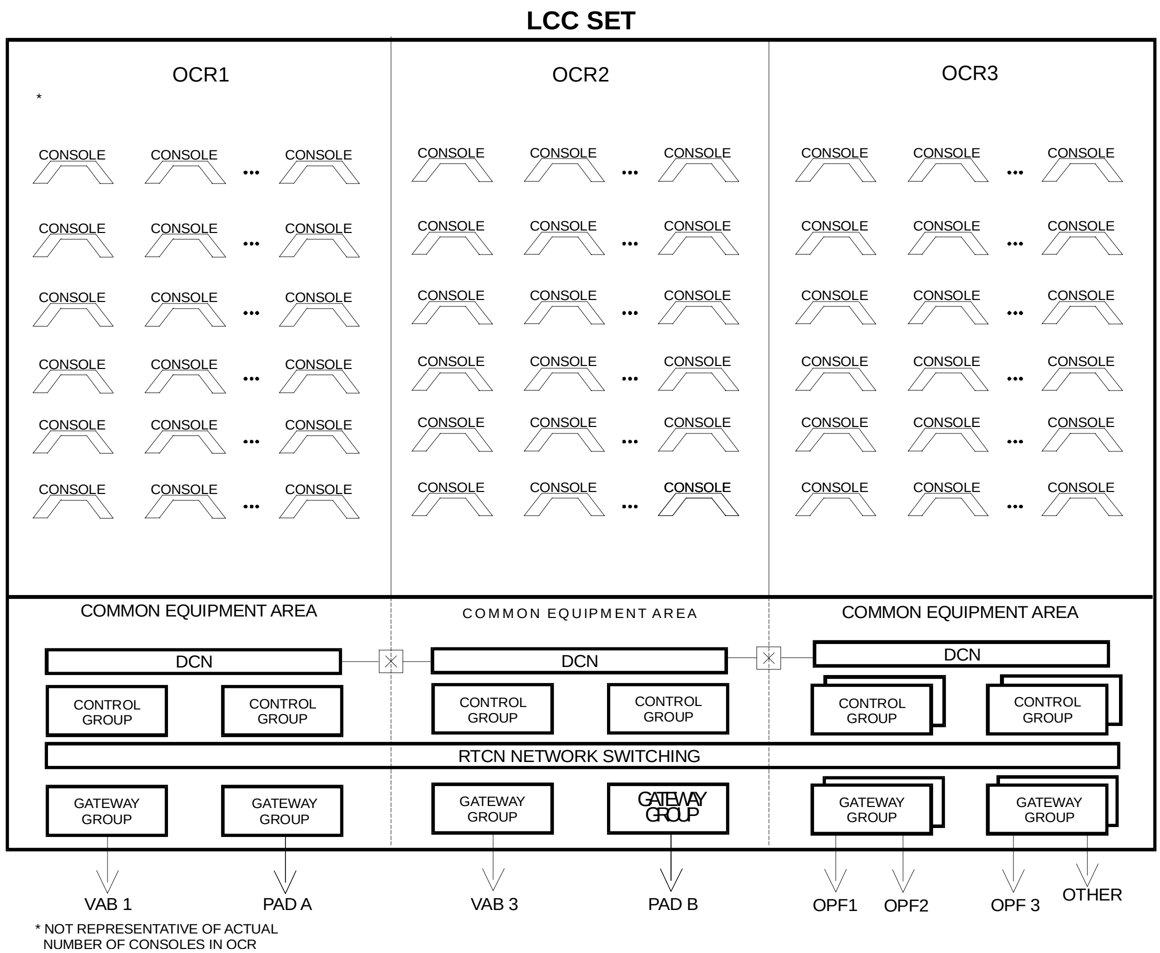


Figure 23 The Launch Control Center Set

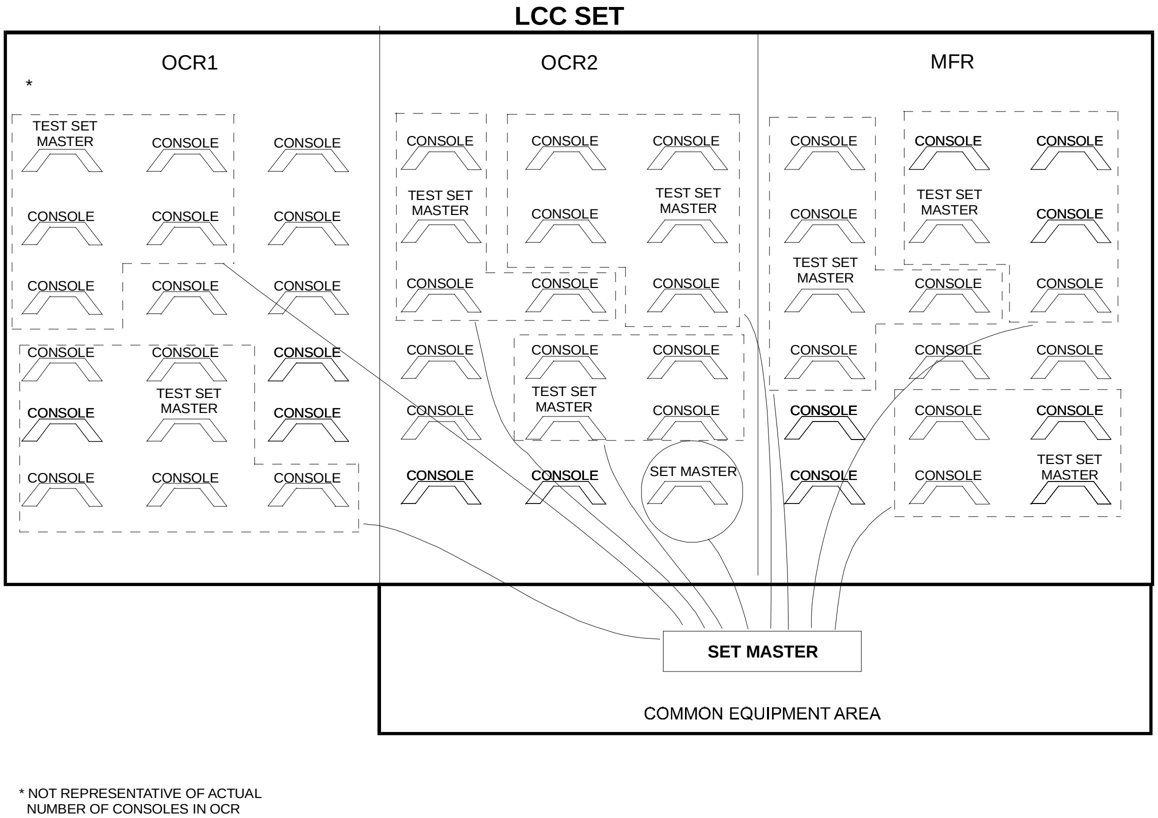


Figure 24 LCC Set Configured to Support Multiple Flows

2.1.5 RTPS External Interfaces

The CLCS RTPS external interfaces are illustrated in Figure 25.

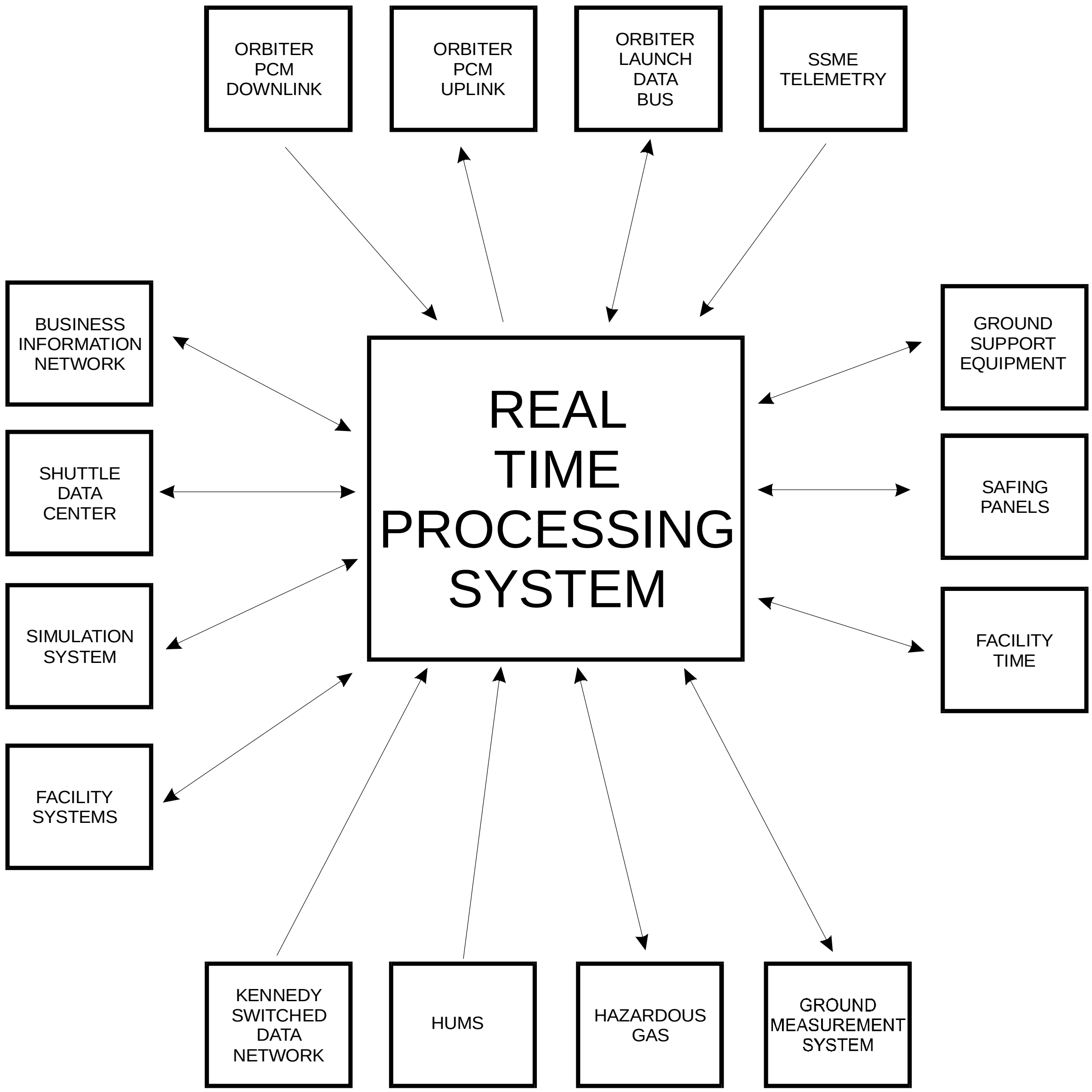


Figure 25 RTPS External Interfaces

The RTPS External Interfaces are itemized in Table 1. This table identifies the Interface Control Documents that describes and controls the signals and data at the interface.

INTERFACE	TYPE	DATA RATE	Interface Control Document
Launch Data Bus	Serial	1 Mbs	SS-P-0002-150N Space Shuttle Computer Program Development Specifications March 1996 SSL LDB Software Interface Requirements Data Bank Services Requirements Specification
PCM Downlink Orbiter OFI	PCM	64 Kbs, 96 Kbs, 128 Kbs, 192 Kbs	SS-P-0002-140T Space Shuttle Computer Program Development Specifications (CPDS) October 1995 SS Downlist/Uplink Software Requirements
PCM Downlink Main Engine	PCM	60 Kbs	

INTERFACE	TYPE	DATA RATE	Interface Control Document
PCM Uplink	PCM	32 Kbs, 72 Kbs, 128 Kbs	SS-P-0002-140T Space Shuttle Computer Program Development Specifications (CPDS) October 1995 SS Downlist/Uplink Software Requirements
Ground Support Equipment	Serial	1 Mbs	
Facility Timing MET	IRIG	1 Mbs	
Safing Panels			
KATS	LDB and PCM		
GMS			
Metro			KSC-DL-3768 Theory of Operations (Metro)
B/U HGDS			
CITE			
EDAMS			
GSE HIM	GDB		
HUMS			
SCAN			
SAIL			
SDS			
SPDMS			
KSDN			
LSDN			
Marshall Space Flight Center			
Johnson Space Center			
Dryden Flight Research Center			
Payload Operations Control Centers			
MER			

Table 1 CLCS External System Interface

2.1.6 CLCS Network Topology

The RTPS processing systems are interconnected with high speed (155 Mbs) Asynchronous Transfer Mode (ATM) networks. The networks are connected through ATM switches. Two networks are used to interconnect the RTPS processing systems; the Real Time Critical Network (RTCN) and the Display and Control Network (DCN) as shown in Figure 26. In addition to these networks, a Business and Information Network provides access to online services from the SDC and the Internet.

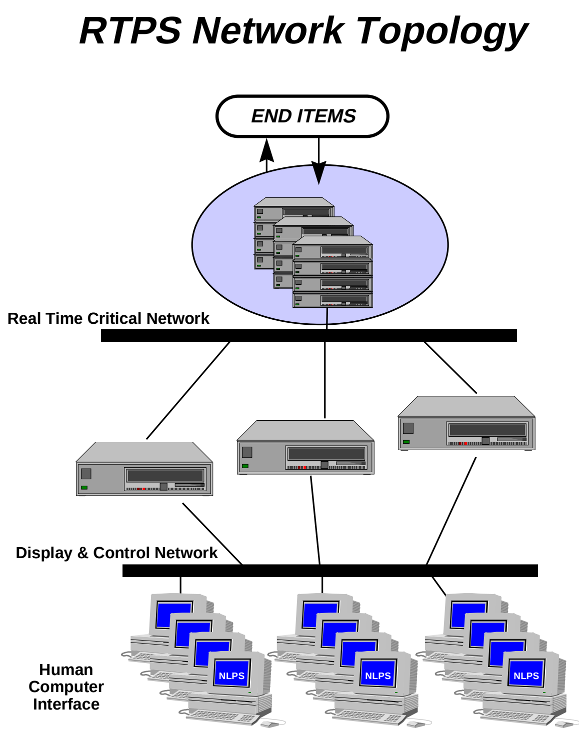


Figure 26 RTPS Network Topology

2.1.6.1 Real Time Critical Network

The Real Time Critical Network (RTCN) provides the transmission means for communications between the Gateway subsystems, the Data Distribution Processing subsystems, the Command and Control Processing subsystems, and the Shuttle Data Center. The RTCN is implemented using an Asynchronous Transfer Mode network operating at 155 Mbs (OC-3). The RTCN implementation is fault tolerant to the extent that no single network failure will cause more than one processing subsystems to be switched from the active to the standby subsystem.

The RTCN implementation in the Launch Control Center Operations Control Rooms (OCR) and Multi-Function Room contains network switching elements that allow flexible connection of RTPS Control Groups and Gateway Groups.

Measurement change data is transferred on the RTCN at the System Synchronous Rate (SSR). The SSR is initially set to be 10 milliseconds. System software establishes a staggered start time from the SSR for each Gateway subsystem to begin transmission of change data. Command data from the CCP's is transmitted asynchronously, on demand, to Gateway subsystems.

A Shuttle Data Center subsystem is attached to the RTCN to record all network traffic. The SDC archives RTPS measurement data and provides retrieval capability to users. During critical operations, the SDC record and playback subsystems are dedicated to the CLCS RTPS test set controlling that critical operation. The SDC also uses the RTCN as the path for loading software and TCID's into the control groups and gateways.

2.1.6.2 Display and Control Network

The Display and Control Network (DCN) provides the transmission means for communications between the Human Computer Interface (HCI) subsystems, the Data Distribution Processing (DDP) subsystems, and the Command and Control Processing (CCP) subsystems. The DCN is implemented using an Asynchronous Transfer Mode (ATM) network operating at 155 Mbs (OC-3).

The DCN implementation in the LCC Operations Control Rooms (OCR) and Multi-Function Room (MFR) contains network switching elements that allow flexible connection of RTPS control groups to HCI subsystems.

The attached SDC records traffic on the DCN. The DCN is also used for retrieving recorded data from the SDC and for loading software and TCID’s into the HCI workstations.

Data transfer on the DCN is synchronized at the Display Synchronous Rate (DSR). The DSR is initially set to be 100 milliseconds.

2.1.6.3 Business Information Network

The Business Information Network (BIN) provides the network connection between external data sources and Human Computer Interface (HCI) subsystems located in the LCC OCR’s and MFR. Information on this network allows system engineers to access:

1. The Shuttle Data Center (SDC)
2. Operational Television (OTV)
3. World Wide Web (WWW)
4. Audio Playback

2.1.6.3.1 Shuttle Data Center

The Business and Information Network (BIN) provides access to the SDC which contains the central repository for engineering documentation, drawings, procedures, and other materials used in vehicle processing and launch operations. The BIN provides access via two console positions at each workstation which are isolated from the RTPS systems and networks.

2.1.6.3.2 Operational Television

Users have access to OTV for viewing and control from a Business and Information position. A Graphical User Interface (GUI) provides the capability to control pan, tilt, zoom, iris, screen capture, sweep and other OTV camera features. A camera sequencing function is provided for viewing. The user has the capability to playback recorded OTV, including slow motion. Screen captured images may be viewed and printed. Up to four OTV cameras images may be displayed on one monitor at a console.

2.1.6.3.3 World Wide Web

The Business and Information network allows users to access the World Wide Web (WWW). Security is provided by a firewall that restricts external access to CLCS information.

2.1.6.3.4 Audio Playback

Audio playback via headsets or speakers is available at the Business Information position at each console.

2.1.6.4 LPS Operational Network

The RTPS provides the Shuttle Data Stream (SDS) to users via the LPS Operational Network (LON). The Consolidated SDS Gateway acquires measurement change data and forwards this information to users in the predefined Shuttle Data Stream format.

2.1.6.5 Shuttle Data Center Interface

The Shuttle Data Center performs the following functions:

1. Configures system and applications software for test operations
2. Maintains and processes measurement and command databases
3. Manages test related documentation
4. Provides measurement recording and playback functions during test operations
5. Provides data archive and retrieval services

The simulation system uses mathematical models of shuttle and GSE systems to simulate hardware systems for verification of test procedures and application software. The simulation system is also used for launch team training.

2.2 Shuttle Data Center

The Shuttle Data Center provides the following support services for the RTPS:

1. Processes the Shuttle Data Tape and the Payload Data Tape
2. Maintains the CLCS Function Designator Data Base
3. Builds Gateway polling, telemetry decommutation, LDB, and PCM Uplink tables
4. Provides the configuration management repository for RTPS system software
5. Provides the configuration management repository for applications software
6. Performs Test Builds
7. Performs System Software Builds
8. Supports loading of RTPS Test Sets
9. Records traffic on RTCN and DCN
10. Provides retrieval services for recorded data

2.2.1 Shuttle Data Center Conceptual Model

This section under construction.

Support Software

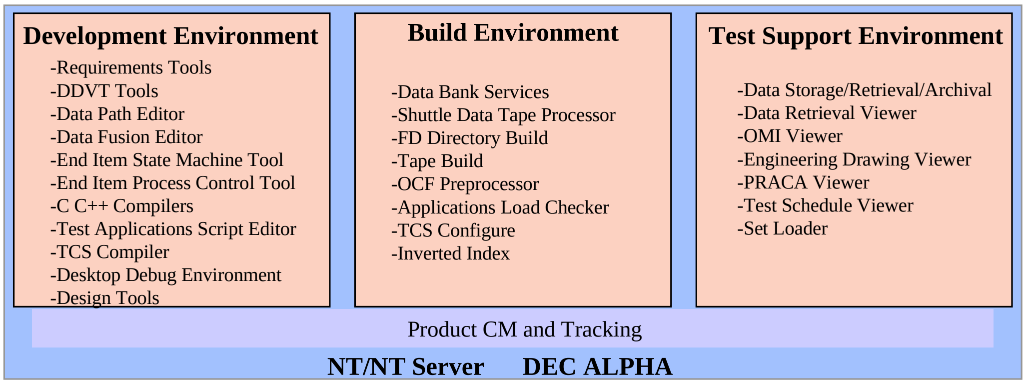


Figure 27 Shuttle Data Center Functions and Services

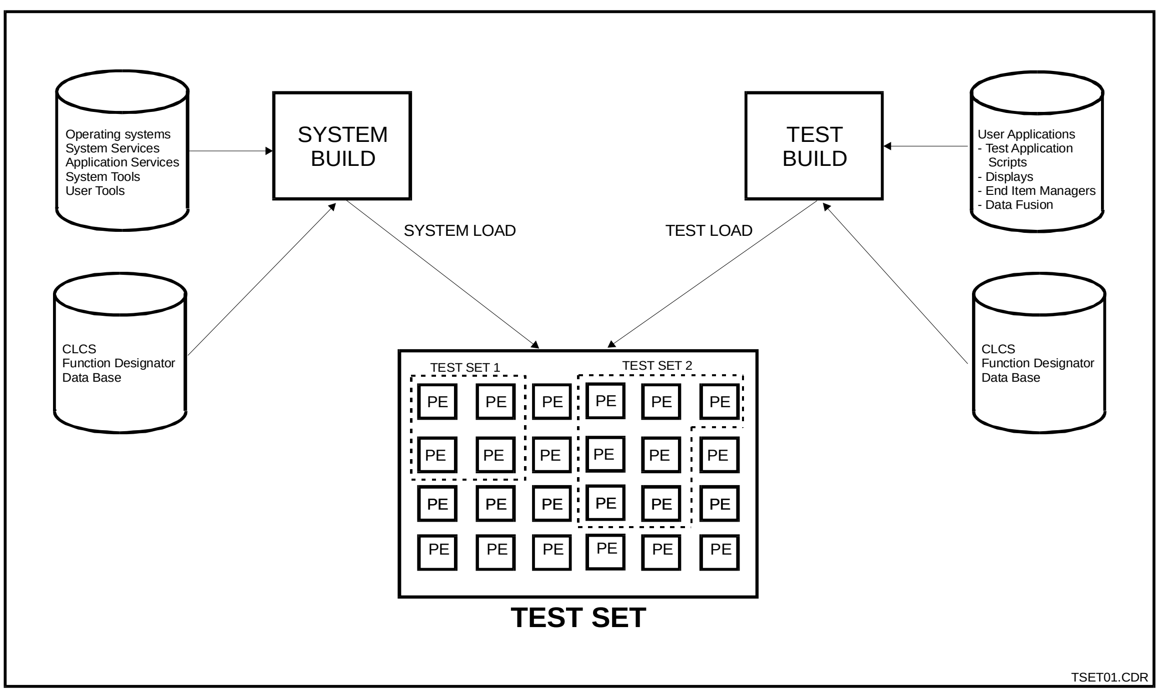


Figure 28 Shuttle Data Center CLCS System Build

The SDC operational environment is distributed across three primary platforms within the CLCS Operational Network (CON). These are:

1. The CLCS Software Development Network (CSDN)
2. The SDC Application Servers (SAS)
3. The CLCS Test Build Server (CTB).

The CSDN is a local area network of HP 9000 Series UNIX workstations running the HP-UX Operating System and interconnected through Ethernet and TCP/IP. These workstations provide support of COTS products used by CLCS Support Software, such as:

1. X-Windows/Motif
2. A database browser
3. Oracle Forms
4. Oracle Reports
5. Oracle SQL*Net
6. Network File System (NFS)
7. NCSA's Mosaic HyperText browser

2.2.1.1 CLCS Software Development Network

The CSDN provides Development Environment capabilities, such as: user accounts, user file space, user-developed tools / scripts, and printing capabilities necessary for supporting day to day operations performed by CLCS Support Software users. CLCS Support Software resident on the CSDN includes Applications Editors and Compilers; DataBank Shuttle Automated Function Executive (DBSAFE) Forms and Reports; and a RDBMS data browser. Interfaces from the Application Editors and Compilers to the DBSAFE CLCS DataBank and the Application Program Library (APL) are also resident on the CSDN.

2.2.1.2 SDC Application Servers

The SDC is a local area network of DEC AlphaServers running DEC UNIX and interconnected through Ethernet and TCP/IP. Several of these servers are designated as application servers (SASs). These servers provide support of COTS products used by CLCS Support Software, such as: X-Windows/Motif, a database browser, Oracle Forms, Oracle Reports, Oracle SQL*Net, and NCSA's Mosaic HyperText browser. The SASs provide Development Environment capabilities, such as: user accounts, user file space, user-developed tools / scripts, and printing capabilities necessary for supporting day to day operations for those users without access to a CSDN workstation. In addition, one or more of the SASs are configured as a HyperText Transfer Protocol (HTTP) Server to support access to CLCS Support Software functions from HyperText browsers such as Mosaic. CLCS Support Software resident on the SASs includes all of the applications resident on the CSDN.

2.2.1.3 CLCS Test Build Server

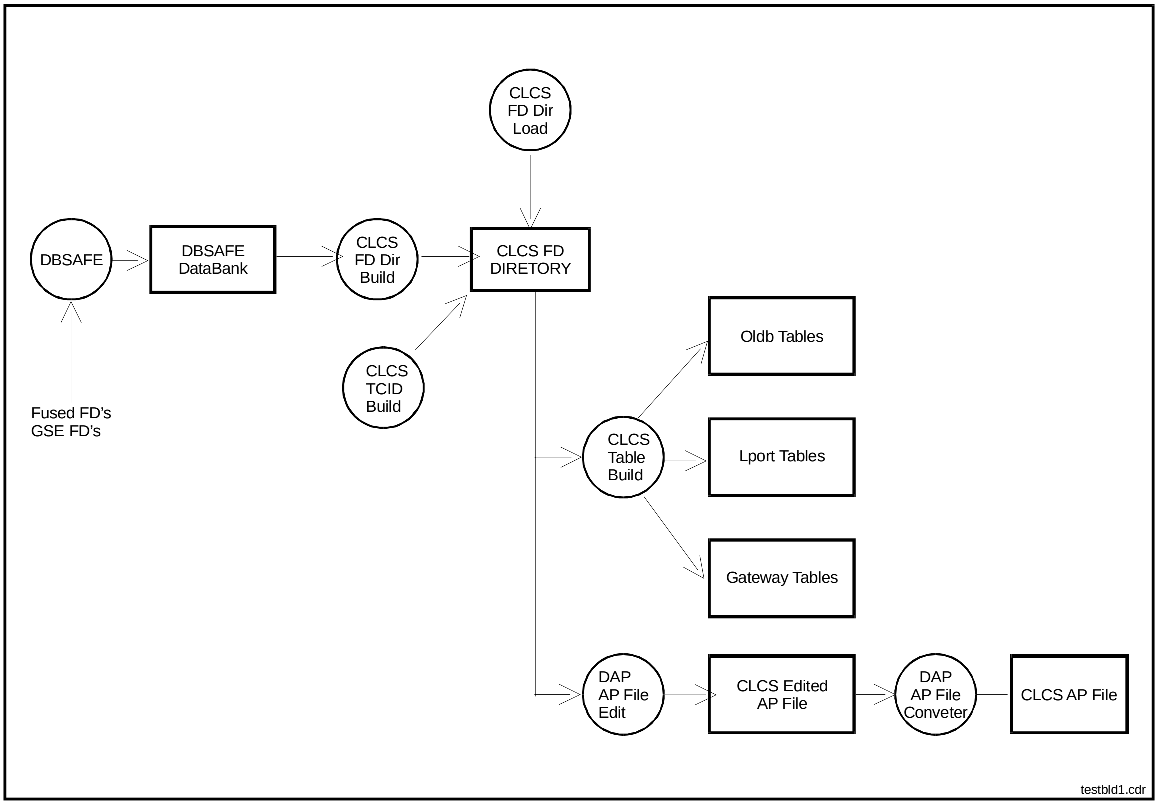
Within the SDC local area network, a DEC AlphaServer 8400 running DEC UNIX is designated as the CTB Server. This server hosts CLCS Test Build Software within the SDC. It hosts the

1. RDBMS engine
2. Reconfiguration Network (RNET)
3. Orbiter Computational Facilities Support Software (OCS) data repository
4. Application Program Library
5. DBSAFE CLCS DataBank
6. Transient TCID databases, directories and files

This server provides a minimum of 12 gigabytes of on-line storage and provides support of COTS products used by CLCS Test Build Software, such as: X-Windows/Motif, Oracle SQL*Net, Oracle RDBMS, Revision Control System (RCS), a HTTP Server, and any required runtime libraries (e.g., Oracle Forms, MicroFocus COBOL, etc.). No user login accounts, user file space or user-developed tools / scripts reside on the CTB Server other than those required for administrative purposes. The CTB Server is dedicated solely to process execution. CLCS Test Build Software resident on the CTB Server includes stored procedures of DBSAFE; APL Maintenance software; all TCID Build and Control software components; and the Common Gateway Interface (CGI) software necessary for initiating processing from HyperText Markup Language (HTML) forms.

2.2.2 Shuttle Data Center Functions and Services

2.2.3 Shuttle Data Center Data Flow



2.2.4 Shuttle Data Center External Interfaces

2.3 Simulation System

The Simulation System uses mathematical models of Shuttle and Ground Support Equipment (GSE) systems to simulate hardware responses to commands from the CLCS RTPS. The Simulation System interfaces with the RTPS via Video Simulation Interface (VSI) processors to produce hardware signals compatible with the Gateway systems hardware interfaces. The Simulation System also provides a network interface that provides emulation of the Gateway systems output on the Real Time Control Network.

Simulating Shuttle and GSE hardware systems, the Simulation System is utilized by software and hardware engineering for testing and verification of CLCS Application Software, test procedures, and CLCS equipment. The Simulation System is also utilized to perform launch team training.

2.3.1 Simulation System Conceptual Model

Simulation Execution Environment		
Simulation Development Tools	**Simulation System Services**	**Test Tools**
•LEX •YACC •Compilers •GUI Development Software •Source Debuger •Source Code Control •Editors •Target (SBC) Development Environment •VSI Development	•Model Converter •Command Language Converter •Model Databank •Flight File Interface •FDDIR Interface •Model System Build •SDC Repository Interface	•DataBus Test Unit •VME Bus Analyzer •Data Simulator/Generator •PCM Test Environment •PMCI Analyzer

Table 2 Simulation System Environment

2.3.2 Simulation System Functions and Services

The Simulation System provides the capability to develop and execute mathematical models of physical hardware systems. The mathematical models consist of algebraic equations that compute analog values and Boolean equations that are evaluated to simulate logic systems. The models accept inputs from hardware stimulus when connected to CLCS gateway interfaces. The models may also be stimulated by users from a simulation console and by Real Time Procedures that execute under control of the simulation system. The simulation system provides special commands to “fail” calculated values to a specific state or value.

In order to facilitate the checkout of CLCS Application Software, test procedures, CLCS equipment, and launch team training functions, the Simulation System provides the following services:

1. Provides a model control user interface connected to a model executing on a Single Board Computer (SBC)
2. Provides the capability to control model execution
3. Provides the capability to modify (i.e., SET, FAIL, and RESET FAIL) model variables (e.g., pseudos, internal variables)
4. Provides the capability to execute model control procedures
5. Provides the capability to monitor and display model variables

2.3.3 Simulation System Data Flow

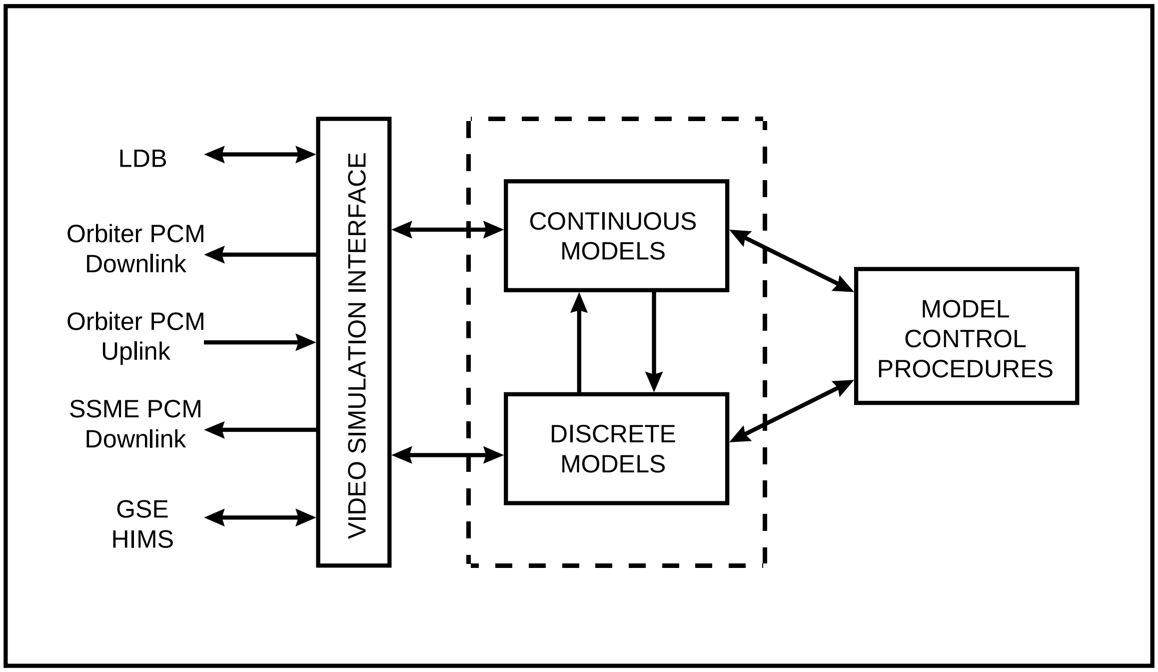


Figure 29 Simulation System Architecture Overview

2.3.4 Simulation System External Interfaces

The Simulation System emulates the following hardware interfaces to RTPS:

1. Launch Data Bus
2. Orbiter PCM Downlink
3. Orbiter PCM Uplink
4. SSME PCM Downlink
5. Ground Support Equipment HIMs

2.4 CLCS Computer Software Configuration Items

The CLCS computer Software Configuration Items are illustrated in Figure 30 CLCS Computer Software Configuration Items.

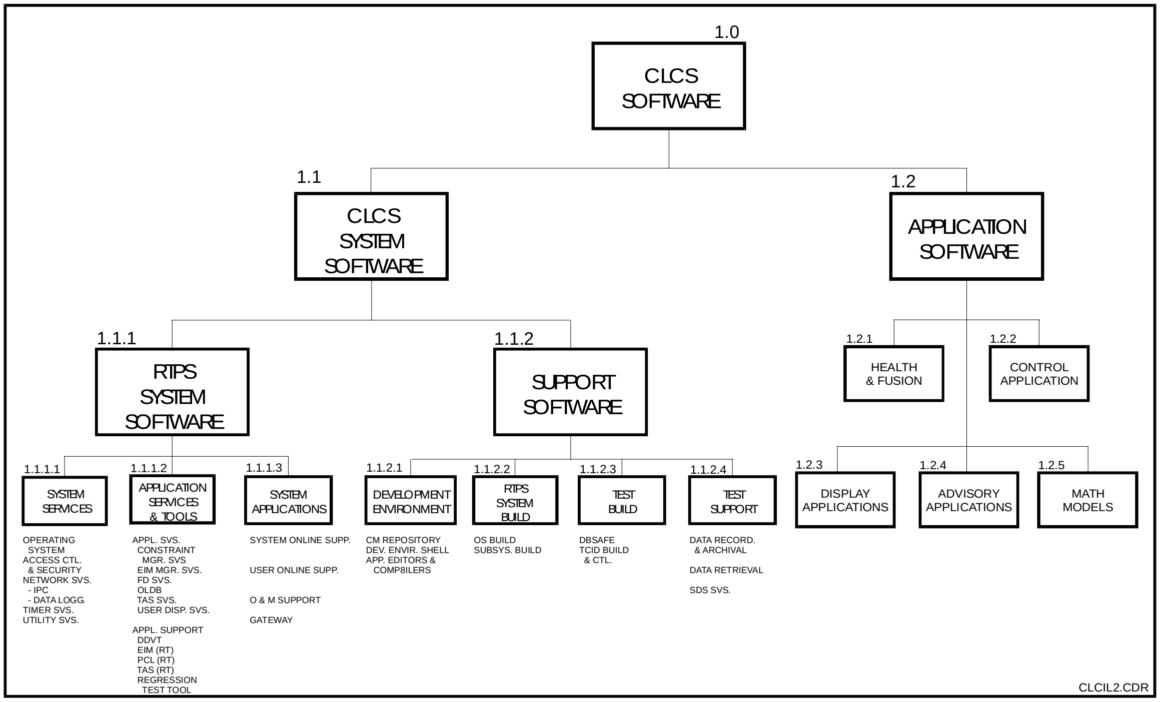


Figure 30 CLCS Computer Software Configuration Items

2.5 Safety, Reliability and Quality Assurance

TBD

2.6 Performance

TBD

3. Real Time Processing System Architecture

3.1 RTPS Software Architecture

The Real Time Processing System (RTPS) software architecture is a layered architecture based on a Commercial-Off-The-Shelf (COTS) POSIX compliant based operating system. The architecture layers are:

1. Operating System
2. System services
3. Application services
4. COTS and System Tools
5. System applications
6. User Applications

The RTPS software architecture is illustrated in Figure 31 RTPS System Software Architecture.

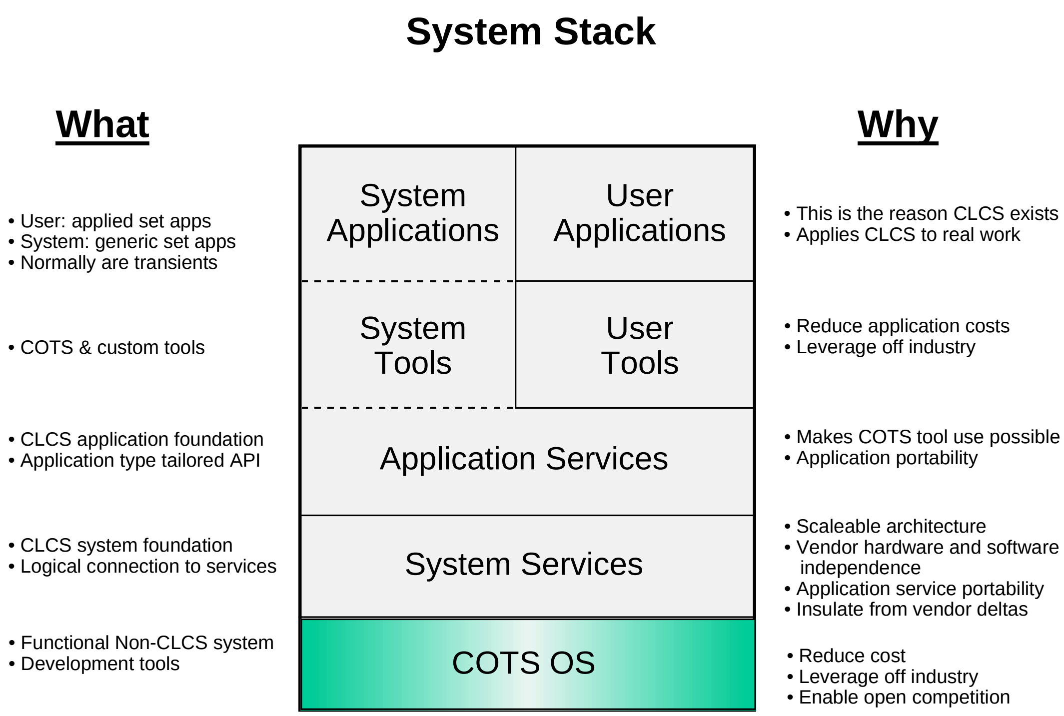


Figure 31 RTPS System Software Architecture

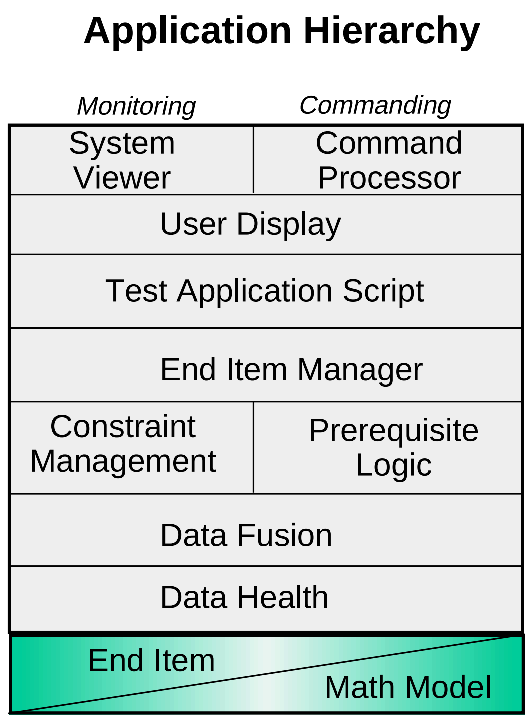


Figure 32 CLCS Application Hierarchy

Application Stack

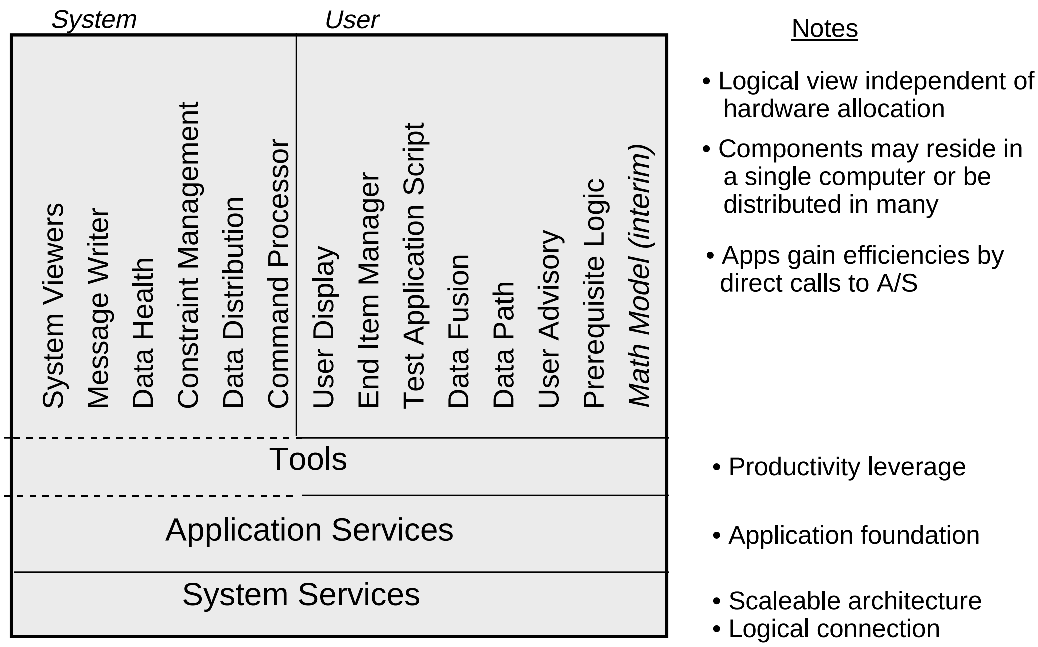


Figure 33 CLCS Application Stack

3.1.1 RTPS Operating Systems

The RTPS operating system for the Data Distribution Processing, Command and Control Processing, and the Human Computer Interface subsystems is supplied by the hardware computer subsystem vendor and is POSIX compliant. The operating system for the Gateway systems is VxWorks provided by Wind River Systems, Alameda, Ca. VxWorks is a POSIX compliant, real-time operating system for embedded computers and is based on a microkernel architecture that supports preemptive and round robin task scheduling.

3.1.2 RTPS System Services

The RTPS System Services are the functions that provide a consistent interface to system applications and user application services. Since the RTPS may be hosted on different hardware platforms, each having a unique version of the operating system, system services provides an interface layer that System applications and user application services use system services to perform functions which may vary from platform to platform. The System Services include:

1. Network services
2. Access Control/Security
3. Inter process communications
4. Data logging
5. Timer services
6. Utility services

3.1.3 RTPS Application Services

The RTPS Application Services are services that provide applications the capability to access hardware end item measurement information, issue commands to the hardware end items, communicate with other applications and display information to CLCS users.
RTPS Application Services include:

1. Constraint Management services
2. End Item Manager Services
3. FD services
4. Inter-Application Communication
5. CLCS Function Designator Data Base
6. Test Application Script Services
7. User Display Services

3.1.4 RTPS Application Support Tools

The RTPS Application Support Tools include:

1. Dynamic Data Visualization Tool (Run Time)
2. End Item Manager (Run Time)
3. Prerequisite Control Logic (Run Time)
4. Test Application Script Manager (Run Time)
5. Regression Test Tool

COTS tools are currently being evaluated for each of the functions and are TBD.

3.1.5 RTPS System Applications

The RTPS System Applications are the applications software components that define the function of a subsystem . The RTPS System Applications include:

1. Data Distribution and Processing
2. User Application Processing
3. System Viewers
4. Command Processor
5. System Control
6. System Diagnostics
7. Common Gateway Services
8. GSE Gateway Services
9. LDB Gateway Services
10. PCM D/L Gateway Services
11. PCM Uplink Gateway Services

3.1.5.1 Operations Configuration Manager

In the CLCS RTPS environment, hardware end item test activities require a CLCS Test Set to be configured with all of the required RTPS hardware, system software and test software. The RTPS Operations Configuration Manager is a software component of RTPS System Control and performs the following functions to initiate and control a test activity:

1. Verifies the RTPS hardware subsystems
2. Loads, verifies and initializes RTPS system software
3. Loads, verifies and initializes the TCID
4. Configures the RTPS hardware subsystems

5. Reconfigures, in real time, RTPS subsystems hardware and software
6. Creates new activities during test operations, in real time
7. Connects, and assures the connection, to the end item hardware under test

Test activities may be predefined or created in real time using a the RTPS Activity Manager. Predefined activities identify the gateway groups, control groups, system software, TCID, and end item hardware involved in the test activity.

Each facility (LCC, CITE, HMF, etc.) has a position allocated for a CLCS Master function. The Master function is the location from which operations personnel configure a Test Set. In the LCC, there is one Master function located in the Common Equipment Area which is used to control all LCC CLCS equipment. An Activity Manager task provides the interface for personnel to configure, test, load, and initialize the subsystems in the Test Set.

The configuration process identifies the processing systems and network connections to build the Test Set. After the Test Set hardware elements are configured, an Operational Readiness Test (ORT) is performed. The ORT identifies any equipment malfunctions or incorrect connections. Once a Test Set is certified operational, a Test Set Master is allocated. The Test Set Master may be any one of the console interfaces within the Test Set. From the Test Set Master, test personnel request software and Test Configuration ID load. System and application software is loaded into the Test Set from the Shuttle Data Center Operations Configuration Manager Server via the local area network.

During test operations, when equipment failures occur, operations personnel use the Activity Manager function to switch from primary to backup hardware elements. In critical operations, when a failure occurs, other hardware elements may be brought into a Test Set to replace failed components. The CLCS operations configuration manager maintains and controls all Test Set hardware and software elements during a test activity. The Test Set configuration may be re-verified during test operations. Non-intrusive hardware testing may be performed. Performance parameters are available to personnel to assist in load balancing the Test Set configuration. System and application software load validity may be test at any time during test activity.

3.2 Gateway Architecture

3.2.1 GSE Gateway Architecture

3.2.1.1 GSE Gateway Functions and Services

3.2.1.2 GSE Gateway Data Flow

3.2.1.3 GSE Gateway Internal Interfaces

3.2.1.4 GSE Gateway Hardware Configuration Items

3.2.1.5 GSE Gateway Computer Software Configuration Items

3.2.2 LDB Gateway Architecture

3.2.2.1 LDB Gateway Functions and Services

3.2.2.2 LDB Gateway Data Flow

3.2.2.3 LDB Gateway Internal Interfaces

3.2.2.4 LDB Gateway Hardware Configuration Items

3.2.2.5 LDB Gateway Computer Software Configuration Items

3.2.3 PCM Downlink Gateway Architecture

3.2.3.1 PCM DOWNLINK Gateway Functions and Services

3.2.3.2 PCM DOWNLINK Gateway Data Flow

3.2.3.3 PCM DOWNLINK Gateway Internal Interfaces

3.2.3.4 PCM DOWNLINK Gateway Hardware Configuration Items

3.2.3.5 PCM DOWNLINK Gateway Computer Software Configuration Items

3.2.4 PCM Uplink Gateway Architecture

3.2.4.1 PCM UPLINK Gateway Functions and Services

3.2.4.2 PCM UPLINK Gateway Data Flow

3.2.4.3 PCM UPLINK Gateway Internal Interfaces

3.2.4.4 PCM UPLINK Gateway Hardware Configuration Items

3.2.4.5 PCM UPLINK Gateway Computer Software Configuration Items

3.2.5 Consolidated SDS Gateway Architecture

The Consolidated SDS Gateway provides an expandable SDS (Shuttle Data Stream) that integrates CCMS SDS Data with data from other data sources such as the GMS and Metro Systems. The new or augmented SDS is PC-Goal Compatible. The Gateway also re-transmits GMS and Metro data to the CLCS Consolidated Systems Gateways.

3.2.5.1 Consolidated SDS Gateway Functions and Services

The Consolidated SDS Gateway acquires Meteorological data from the LC-39 Meteorological Data System and retransmit the raw Meteorological data to the Consolidated Systems Gateway. The floating point Meteorological data is converted to GPC Floating Point to perform calculations that determine surface ice buildup on the External Tank.

The CSDS Gateway also acquires Ground Measurement System (GMS) data from the GMS servers. The floating point GMS data is converted to GPC Floating Point and user to perform GMS calibration.
The CSDS Gateway merges LC-39 Meteorological System data and GMS data into the each high speed Shuttle Data Streams.

3.2.5.2 Consolidated SDS Gateway Data Flow

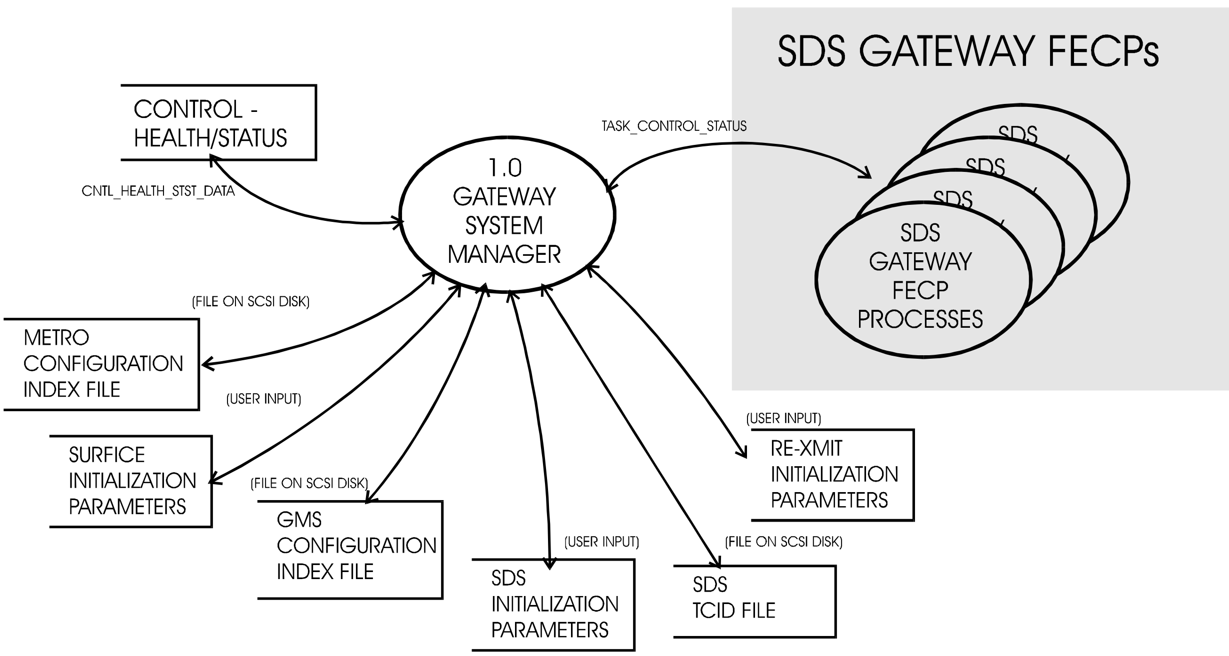


Figure 34 Gateway System Manager Data Flow Diagram

3.2.5.3 Consolidated Gateway External Interfaces

The CSDS Gateway uses the following externally provided data files:

1. Metro Configuration Index File
2. Surfice Initialization Parameters
3. GMS Configuration Index File
4. SDS Initialization Parameters
5. SDS TCID File
6. Re-Transmit Parameters

3.2.5.4 Consolidated SDS Gateway Hardware Configuration Items

The Consolidated SDS Gateway consists of a 19 inch equipment rack that houses the VME Telemetry Processor and other associated support hardware. **Error! Reference source not found.** lists the Hardware Configuration Items (HWCIs) that comprise the Consolidated SDS Gateway:

HWCI Name	Implementation	GOTS/COTS/Custom
Rack	CORE	GOTS
VME Telemetry Processor		
VME Chassis	CORE	GOTS
GCP	MVME 167	COTS
FEPC	MVME 167	COTS
Transition Modules	MVME 712	COTS
IRIG B Decoder	TRUETIME VME SG-2	COTS
Touch Screen	CORE	GOTS
Patch Panel	Emerald Computer	GOTS
Communications Server	Datability VCP 1000	COTS
Modems	PENRIL	COTS
Power Distribution Panel	CORE	GOTS

Table 3 Consolidated SDS Gateway HWCI's

3.2.5.5 Consolidated SDS Gateway Software Configuration Items

The Consolidated SDS Gateway Service CSCI is composed of CSCs written in the C programming language and run under the VxWorks Operating System. The context diagram is shown in Figure 5.

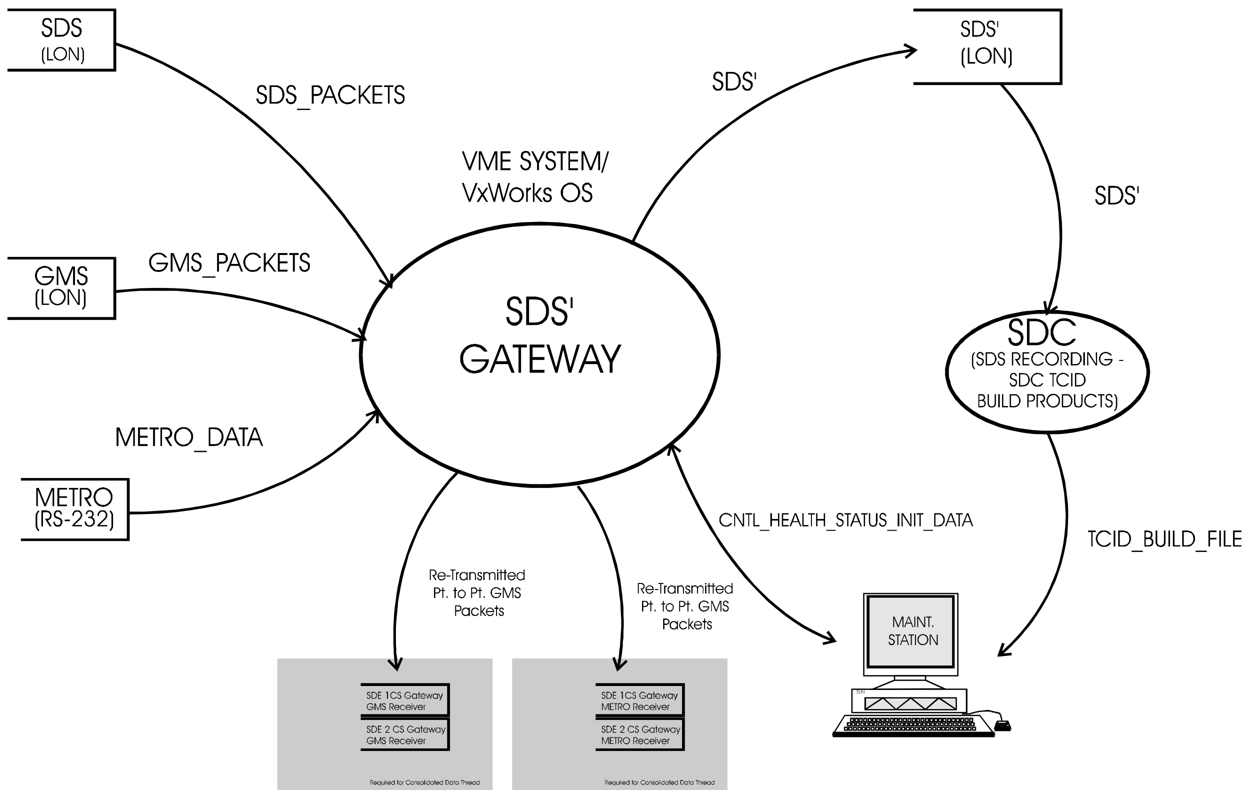


Figure 35 CSDS Gateway Context Diagram

The Consolidated SDS Gateway Services CSCI is composed of several CSCs running on both the Gateway Control Processor and the four Front End Control Processors. A summary is shown in Table 2.

CSC	Function	Processor
Gateway System Manager	System initialization, Control, Health, Status, Hard disc access.	GCP
Metro	Process Data from Pad Meteorological System	FEPC
GMS	Process Data from the Ground Measurement System	FEPC
Retransmit	Re-send raw Metro and GMS Data to Consolidated Systems Gateways	FEPC
Augment SDS	Process SDS and Generate SDS' with GMS and Metro Measurements.	FEPC

Table 4 Consolidated SDS Gateway CSC's

3.2.5.6 Consolidated SDS Gateway Inventory Items

TBD

3.2.6 Consolidated Systems Gateway Architecture (Redstone)

3.2.7 Instrumentation Gateway Architecture (TBD)

3.2.8 IUHM Gateway Architecture (TBD)

3.2.9 Consolidated Systems Gateway Architecture (Redstone)

3.2.10 Instrumentation Gateway Architecture (TBD)

3.2.11 IUHM Gateway Architecture (TBD)

3.3 Data Distribution Processing Architecture

The Data Distribution Processing architecture spans across three types of platforms, which include the Data Distribution Processor (DDP), the Command and Control Processor (CCP) and the Human Computer Interface (HCI) workstations, connected via two ATM networks. Hardware measurement Change Data (function designators) is received from the Gateways at the DDP via the Real Time Critical Network (RTCN), processed at the DDP, re-distributed to the CCP via the RTCN at the System Synchronous Rate (SSR), and to the HCI's via the Display and Control Network (DCN) at the Display Synchronous Rate (DSR). Application derived function designators are receive from the CCP via the RTCN and process in the same manner as hardware measurements.

3.3.1 Data Distribution Processing Functions and Services

Data Distribution consists of the following functions:

- Network Reader
- Data Collection
- Data Fusion Processing
- Data Health Processing
- Data Constraint Processing
- CVT Update
- Function designator queuing
- Packet Building
- Network Writer

The following functions are applied at the DDP:

- Network Reader
- Data Collection
- Data Fusion Processing
- Data Health Processing
- Data Constraint Processing
- CVT Update
- Function Designator Queuing
- Packet Building
- Network Writer

The following functions are applied at the CCP:

- Network Reader
- CVT Update
- Packet Building
- Function Designator Queuing
- Network Writer

The following functions are applied at the HCI:

- Network Reader
- CVT Update
- Function Designator Queuing

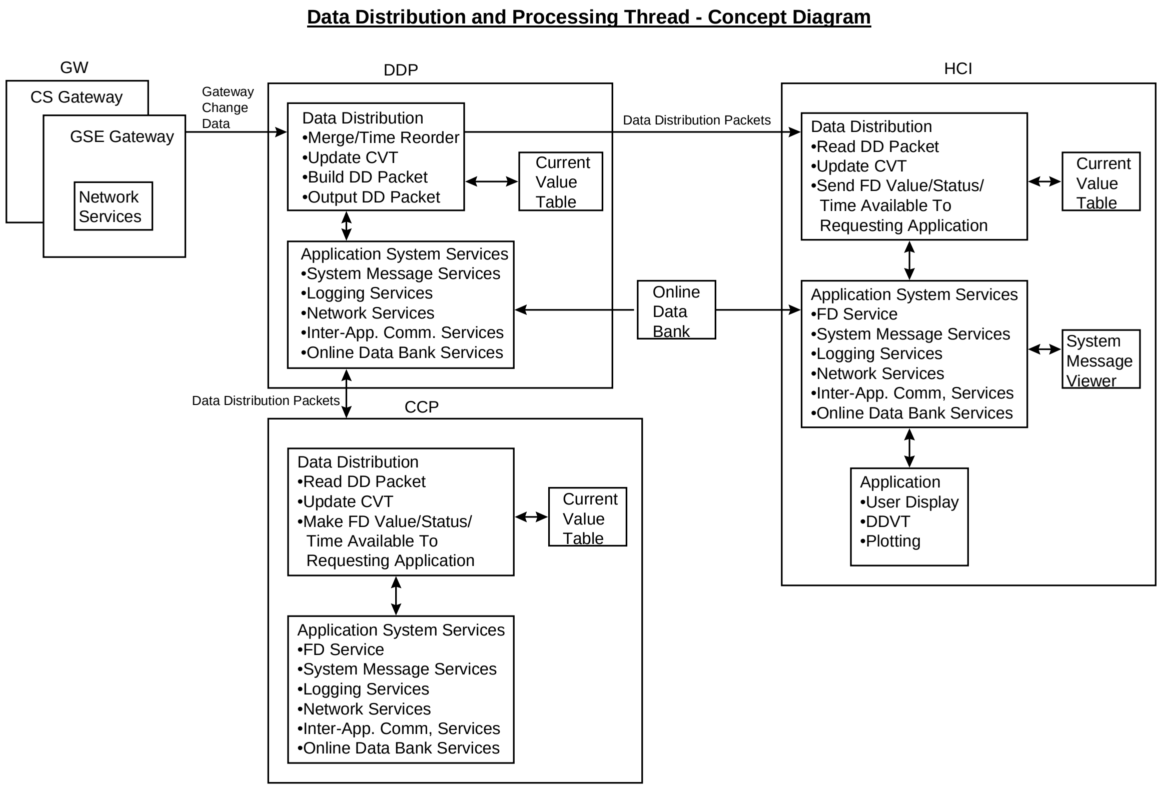


Figure 36 Data Distribution Processing Architecture

3.3.1.1 Network Reader

The Network Reader receives data packets from the network via a Network Services API. At the DDP, the following data packets are read from the RTCN:

- Change data from multiple gateways
- Application derived measurements

At the CCP, Data Distribution packets are received from the RTCN. At the HCI's, Data Distribution packets are received from the DCN.

3.3.1.2 Data Collection

The data collection function merges data received from different packets and time-orders the data.

3.3.1.3 Data Fusion Processing

Data Fusion involves computations using constants, measurement values, health values or other fusion values. The result of the computation is a value which has a type equal to the data fusion Function Designator (FD) as defined in the CLCS Databank. Each fusion function designator found in the databank has the same attributes that any other function designator of the same type would have with the exception that a Fusion function designator does not have a hardware record but does have a fusion algorithm table associated with it. The user may use the CLCS Data Fusion Editor to aid in the input of the fusion algorithm and associated information.

The Data Fusion function performs computations on the measurement values in the DDP based on fusion algorithms and associated information defined in an off-line generated Fusion Algorithm table.

3.3.1.4 Data Health Processing

Data Health is based upon a number of parameters, some of which may be external to the system. Data Health is the term applied to the integrity of a function designator (FD) value which is being distributed from a CLCS subsystem. It consists of a group of flags which are associated with every function designator. Each of the flags is "owned" by different processes within the CLCS system. For example, flags that deal with the decommutation of the data from its source are "owned" by processes in the CLCS Gateway. Other flags are owned by processes which correlate various data to determine additional "health" information about one or more Function Designators. This information is available to all CLCS processes which utilize function designator data.

The Data Health Function applies Data Health information to the measurement data at the DDP.

3.3.1.5 Data Constraint Processing

The Data Constraint Function runs on the DDP. It monitors a set of function designator values and data health changes, compares the actual function designator values to the expected values, and sends event notification to a specific set of applications when certain criteria is met. The expected values, the destination applications and the criteria for event notification, are all defined in an off-line generated Constraint Table.

3.3.1.6 Current Value Table Update

The Current Value Table (CVT) resides on the DDP, CCP and HCI. It contains the current value of all function designators that have been received and processed at each platform. After every cycle of data is received and processed at each platform, the Current Value Table Update function updates the CVT with the new value or status of the changed function designators.

3.3.1.7 Function Designator Queuing

The Function Designator Queuing Function supports placing data in a specific queue and pulling data from a specific queue. The term queuing is a broad term which applies to a physical queue (e.g. a message queue) or a logical queue (e.g. linked list in shared memory). For Data Distribution processing, queues are used for transfer of function designators within the same node.

At the DDP, the function designator Queuing Function is used to pass function designators to Data Fusion processing, Data Health processing, Data Constraint processing.

At the CCP, the Function Designator Queuing Function is used to transfer function designators to System and User Applications, or for the User Application to place User Application Derived function designators into the Data Distribution Output queue.

At the CCP, the Function Designator Queuing Function is also used to pull User Application Derived function designators from a designated queue, build a Data Distribution packet, and output to the RTCN.

At the HCI, the Function Designator Queuing Function is used to transfer function designators to System and User Applications.

3.3.1.8 Packet Building Function

The Packet Building Function collects the data to be output to the Network and places the data in a buffer in a specified Payload Packet format, complete with header information. This function is used both at the DDP and the CCP to build Data Distribution packets. Data packets on the RTCN are transmitted at the System Synchronous Rate (SSR), data packets on the Display and Control Network are transmitted at the Display Synchronous Rate (DSR)

3.3.1.9 Network Writer

The Network Writer Function sends the Payload Packet to the Network via a Network Services API.
This function is used both at the DDP and the CCP to output Data Distribution packets to the RTCN and/or DCN.

3.3.2 Data Distribution Processing Data Flow

TBD

3.3.3 Data Distribution Internal Interface

Table 5 identifies internal data interface and source of data within the DDP

Data Type	Source	Description
Change Data	Data Collection	The Change data collected is sent to Data Fusion Processing, Data Health Processing and Data Constraint Processing.
Fused Data	Data Fusion	The fused data is sent to the CVT and also to be built into the Data Distribution packet to be output to the RTCN and DCN.
Updated Health	Data Health	The updated health is sent to the CVT.

Table 5 Data Distribution Processing Internal Interfaces

Table 6 identifies internal data interface and source of data between the DDP and the CCP

Data Type	Source	Description
Change Data/Fused Data	DDP	Change data and Fused data from the DDP is received at the CCP and stored in the CVT.

Table 6 DDP to CCP Interfaces

Table 7 identifies internal data interface and source of data between the DDP and the HCI

Data Type	Source	Description
Change Data/Fused Data	DDP	Change data and Fused data from the DDP is received at the CCP and stored in the CVT.

Table 7 DDP to HCI Interfaces

3.3.4 Data Distribution Processing Hardware Configuration Items

Data Distribution Processing spans across three hardware platforms:

- DDP (SGI Origin)
- CCP (SGI Origin)
- HCI (SGI O2)

3.3.4.1 Data Distribution Processing Computer Software Configuration Items

The Data Distribution Processing Computer Software Configuration Item (CSCI) is made up of 4 Computer Software Components (CSC):

- The Data Distribution CSC consists of software that performs the following functions:
    - Collect change data from multiple gateways
    - Merge the function designators received from the gateways in a chronological order
    - Supply change data to be processed by the Data Fusion CSC, Data Health CSC, and the Data Constraint CSC.
    - Incorporate the change data, fused data and updated health into a Data Distribution packet
    - Update the DDP CVT with current data
    - Output the packet to the RTCN at System Synchronous rate and to the DCN at Display Synchronous rate.
    - Receive Data Distribution packets at the CCP and HCI
    - Update the CCP/HCI CVT with current data received
- The Data Fusion CSC consists of software that performs the following functions:
    - Apply fusion logic on change data passed from the Data Distribution CSC, based on algorithms defined in the Fusion Table.
    - Pass the fused data to the Data Distribution CSC for output to the RTCN and DCN.
- The Data Health CSC consists of software that performs the following functions:
    - Apply Health to the change data passed from the Data Distribution CSC, based on algorithms defined in the Health Algorithm Table.
    - Pass the updated health to the Data Distribution CSC for output to the RTCN and DCN.
- The Data Constraint CSC consists of software that performs the following functions:
    - Apply constraint logic to the change data passed from the Data Distribution CSC, based on criteria defined in the Constraint Table
    - Send event notification to the target application when the criteria is met. The target application is also defined in the Constraint Table

3.4 Command and Control Processing Architecture

The Command and Control Processing function authenticates and transmits commands, and provides notification of command and measurements events to CCP applications. The CCP also supports the Application Programming Interfaces (API) that provide applications with access to function designator information and commanding functions.

3.4.1 Command and Control Processing Functions and Services

The Command and Control Processing functions consist of:

1. Command Management
2. Event Management
3. Applications Programming Interface Services

3.4.1.1 Command Management

Command management processes application and keyboard command requests. Function designator ownership is verified prior to issuing commands to end items. Commands that have registered prerequisite control logic sequences, are queued for the prerequisite sequence. Command requests are then transmitted to the gateway responsible for the function designator. Success or failure indicators are returned to the originating process along with reason codes.

3.4.1.2 Event Management

Event management controls the registration and notification of system and application events. The event manager accepts requests from system and user applications to be notified of the occurrence of specific system events. Incoming events are routed to the event manager. If an application has registered for that event, a signal is sent to the application and a message placed on the applications event queue. Applications may specify and cancel event notification and conditionally inhibit event notification. Events can be triggered on the occurrence of user input events, function designator constraint violations, timer expiration, or a specified system time.

3.4.1.3 Application API Services

Application API services provides an interface for application programs to access system services. These services include: function designator processing, subsystem control and status, human computer interface control, application control, error handling, timer services, messaging services, and logging services.

3.4.2 Command and Control Processing Flow

Figure 1 illustrates the command and control processing flow.

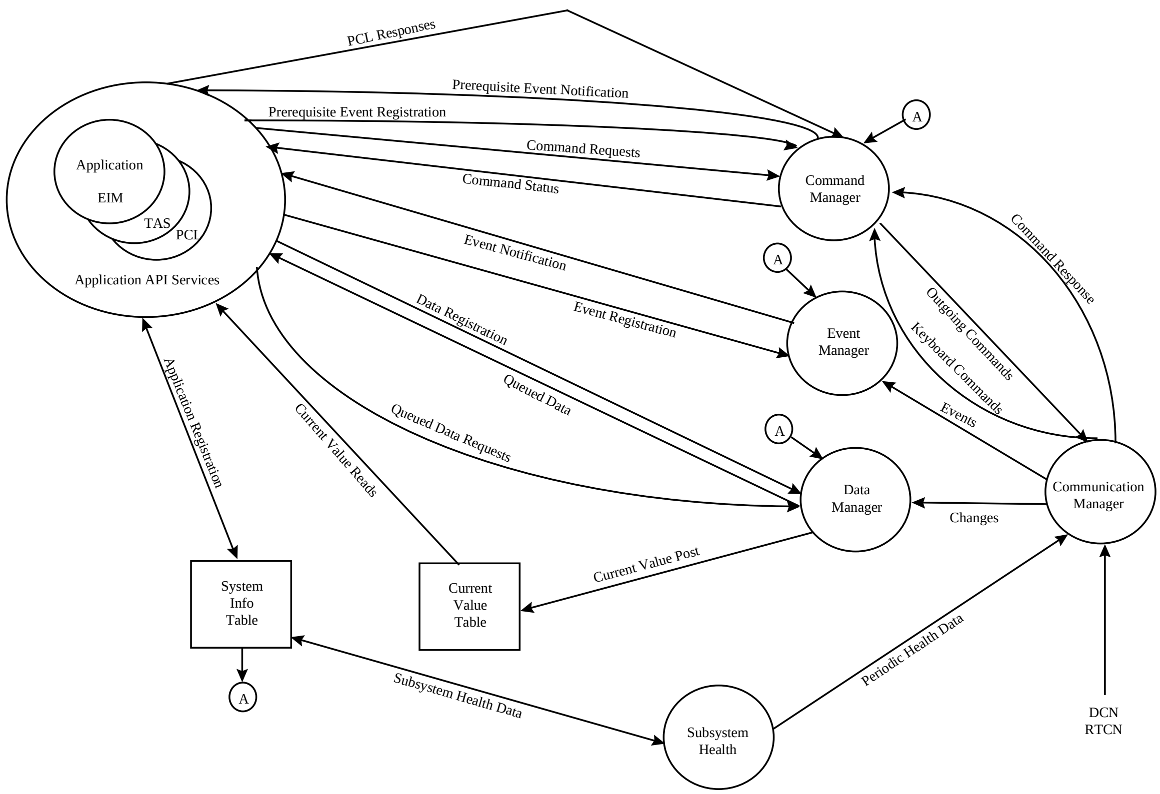


Figure 37 Command and Control Processing Flow

3.4.3 Command and Control Processing Internal Interfaces

The following sections describe the type of data interfaces used by the CCP functions.

3.4.3.1 Command Management Interface

The following table identifies the data interface and source of data for the command management function;

Data Type	Source	Description
Keyboard Command Message	Communication Manager	This message is received when a keyboard command has been issued at the HCI.
Application Command Request	Application Programs	This message is received when an application wishes to issue a command.
Command Responses	Communications Manager	This message is received in response to a command issuance and contains the return status of the command.
Prerequisite Event Registration	PCL Application	Message received from a PCL application identifying a command which should trigger a prerequisite event notification to the PCL application.
PCL Response	PCL Application	A message sent to command management in response to a prerequisite event notification. This message indicates whether a command should be allowed or blocked based on prerequisite logic.

Table 8 Command Management Interfaces

3.4.3.2 Event Management Interface

The following table identifies the data interface and source of data for the event management function;

Data Type	Source	Description
System Information	System Information Table	Data concerning communications information for CCP processes including process ID's, application types, health, etc.
Event Registration Message	Application Programs	Message received from an application identifying an event for which the application wishes to receive notification.
Event Control Messages	Application Programs	Messages received from applications to control the processing of specific events.
Event Notification Message	Communications Manager	Message received from Communications Management when a system event is detected.

Table 9 Event Management Interfaces

3.4.3.3 Subsystem Health Interface

The following table identifies the data interface and source of data for the subsystem health management function;

Data Type	Source	Description
Subsystem Health Data	System Information Table	Information read from the system information table concerning the health and status of processing running within the CCP.

Table 10 Subsystem Health Management Interfaces

3.4.3.4 Application API Interface

The following table identifies the data interface and source of data for the application API services function;

Data Type	Source	Description
Prerequisite Event Notification	Command Manager	Message received when the command manager receives a request to send a command that has a related prerequisite sequence application. PCL applications only.
Command Status	Command Management	A message received in response to a request to issue a command. The message indicates the success or failure status of a command.
Event Notification	Event Management	A message and interrupt signal received when an event is detected for which the application has indicated it should receive notification.
Queued Data	Data Management	A message which contains the current value, time and status for a particular FD change event.
Current FD Values	Current Value Table	The current value of an FD as read from the current value table.
Application Information	System Information Table	Information used by applications to identify other applications and services in the system. Used for things like determining inter process communication paths.

Table 11 API Interfaces

3.4.4 Command and Control Processing Software Configuration Items

3.4.4.1 Command Management CSCI

Figure 38 illustrates the CSCI breakdown for the Command Management CSCI in the CCP.

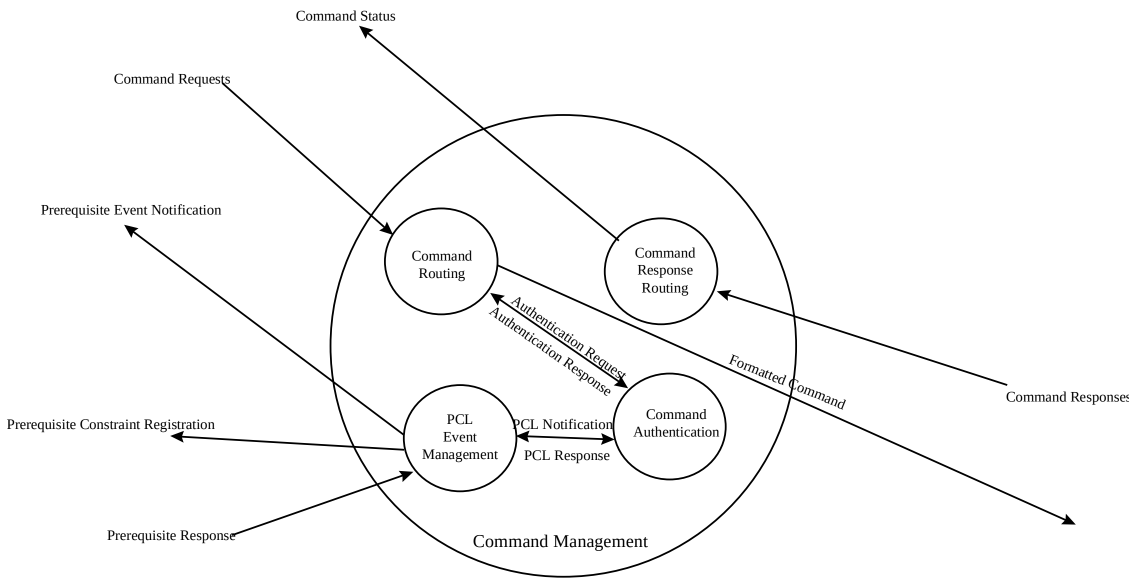


Figure 38 Command Management CSCI

3.4.4.1.1 Command Authentication

Command authentication is performed on all application or user console command requests. Command authentication verifies format, data type compatibility, permission, and command enabled prior to issuing a command. Command authentication returns an acknowledgment to the command router indicating whether or not the command has been cleared for issuance.

3.4.4.1.2 Command Routing

Command routing accepts command requests from applications or user consoles, authenticates the command, determines the destination, formats, and transmits the command to its destination. If the command has an associated prerequisite control logic sequence, the destination is PCL event management. An event will be posted on the Command Response event queue.

3.4.4.1.3 PCL Event Management

Prerequisite control logic (PCL) event management provides the following PCL support:

- Accepts messages from control logic applications identifying the command events for which the PCL application requires notification.
- Maintains a list of the commands which trigger PCL application notification.
- Accepts messages from command authentication identifying command requests.
- Notifies PCL applications when commands are being issued for which there is a prerequisite application.
- Receives messages from PCL applications indicating whether a command is cleared or should be blocked.

Once these functions have been performed PCL event management returns a message to command authentication indicating whether a command is cleared or blocked by prerequisite logic.

3.4.4.1.4 Command Response Processing

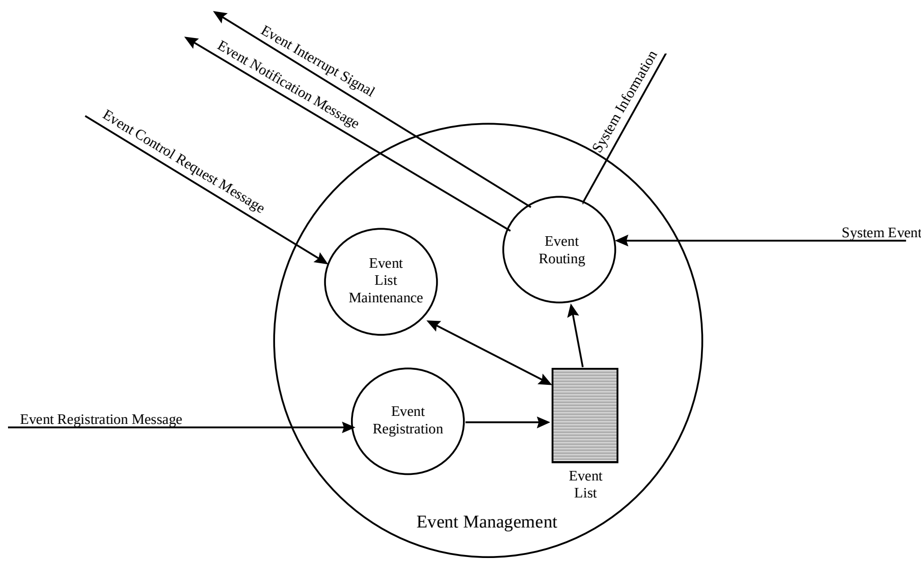
Command response processing accepts responses to commands issued by application or keyboard users and returns the status to the requester. Command Response Processing:

- Verifies that the command response is for an outstanding command issued by the CCP.
- Processes the outstanding command list to determine if commands have timed out.
- Removes the command from the outstanding command list.

Once these functions have been performed a message is sent to the requesting application or keyboard requester indicating the status of the command.

3.4.4.1.5 Event Management CSCI

Figure 3 shows the CSCI breakdown for the event management CSCI in the CCP.



3.4.4.1.6 Event Registration Management

Event registration management maintains the notification event list for application programs. The activities performed by this function are:

- Accept event notification registration messages from application programs indicating what system events should cause notification of the application to occur.
- Verify event registration messages are properly formatted and contain valid event identifiers.
- Maintain an event notification list indicating which events have associated notification requirements.

After these functions are performed an event notification element is placed on the event list which indicates under which circumstances an application should be notified of the occurrence of an event. Event notification can be specified for HCI events, function designator constraint violation events, timer expiration events, or system time arrival.

3.4.4.1.7 Event List Maintenance

Event list maintenance maintains the entries in the event notification list. These services include:

- Accepting messages from application programs to cancel, inhibit, or activate event notification events.
- Verify the requested event is valid and in the current event list structure.
- Modify or remove the event list entry to reflect the requested change.

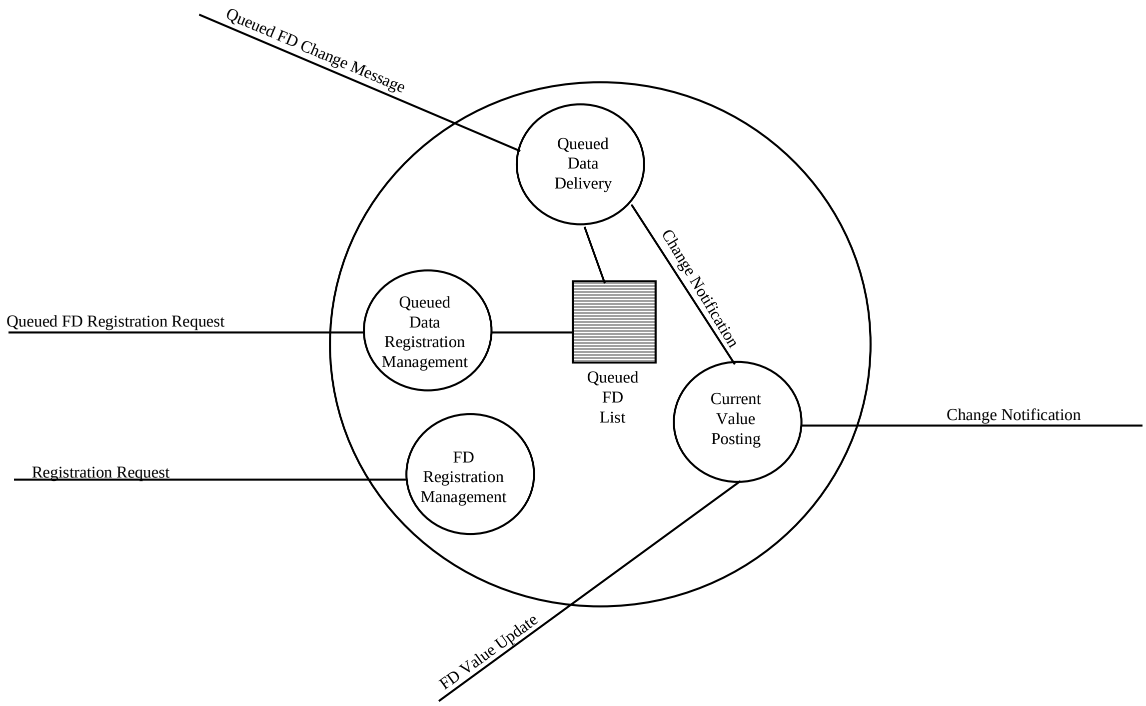
Upon completion of the above functions a message indicating the status of the change request id returned to the calling application.

3.4.4.1.8 Event Routing

Event routing processes system events and directs them to the required application. Event routing receives incoming system event messages, searches the event list for the application that has requested notification, places on the applications event queue, and signals the application that an event message is pending on the application's event queue.

3.4.4.2 Data Management CSCI

Figure 3 shows the CSCI breakdown for the data management CSCI.



3.4.4.2.1 Function Designator Registration Management

3.4.4.2.2 Queued Data Registration Management

The queued data registration manager is responsible for registering multi-sample function designator delivery requests for applications. The functions involved in this are:

- Accepting queued data request messages from applications indicating function designators for which queued data service is requested.
- Verifying system limits on queued data have not been exceeded.
- Placing an entry in the queued function designator list indicating a queued service is requested.

After these functions are performed a queued data request entry exists on the queued function designator list.

3.4.4.2.3 Queued Data Delivery

Queued data delivery is responsible for routing all samples of change data for queued function designator's to the application which has requested the data. The functions involved in this are:

- Accepting function designator change notifications from the function designator value posting function.
- Searching the queued function designator list to determine if an application has requested queued data service for the function designator.
- If a list entry exists, format an function designator change message and place it on the applications function designator change queue.

When these functions are complete queued data delivery signals the application that a data change message is pending on the applications data change queue.

3.4.4.3 Application Services API CSCI

The application services API provides the application access to the functions of the other CCP CSCI's.

3.5 Human Computer Interface Architecture

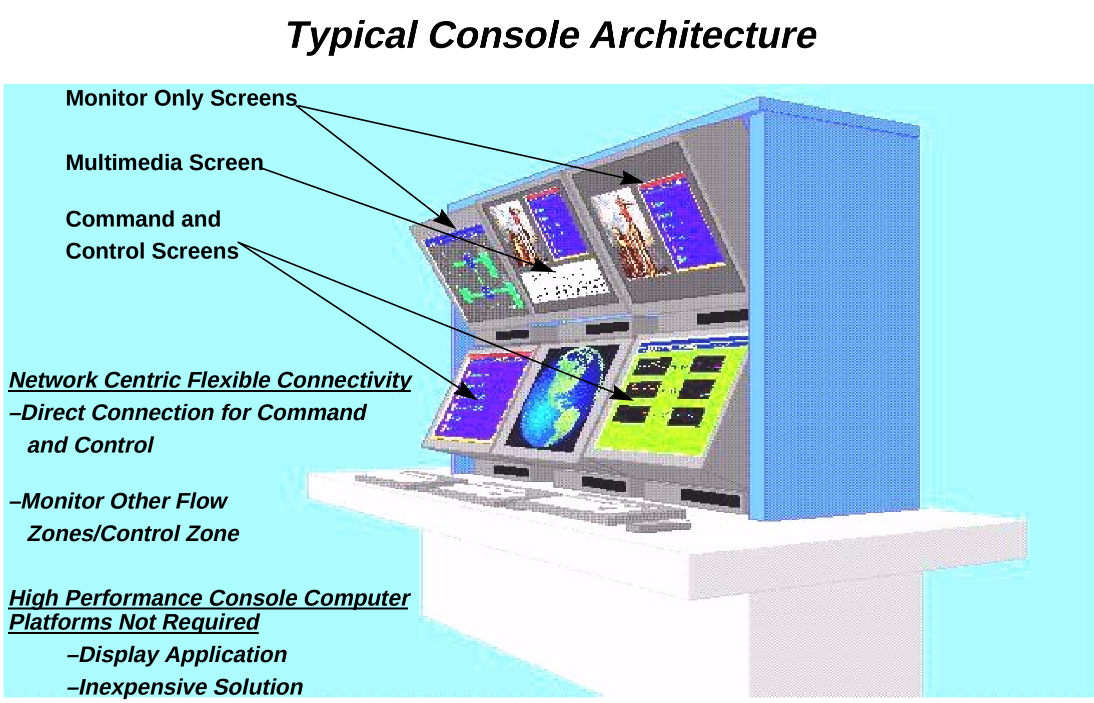


Figure 39 RTPS Console Architecture

3.5.1 Human Computer Interface Functions and Services

3.5.2 Human Computer Interface Data Flow

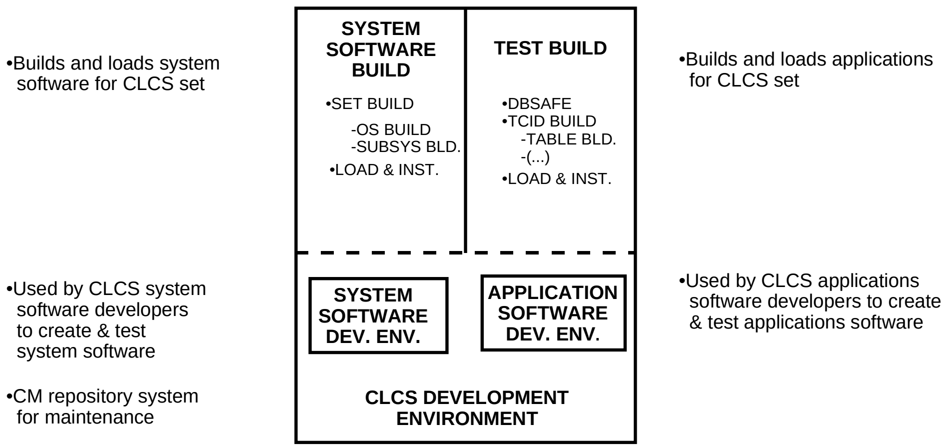
3.5.3 Human Computer Interface Internal Interfaces

3.5.4 Human Computer Interface Hardware Configuration Items

3.5.5 Human Computer Interface Computer Software Configuration Items

4. Shuttle Data Center Architecture

CLCS BUILD/DEVELOPMENT SUPPORT SOFTWARE



4.1 System Software Build Architecture

4.2 Test Build Architecture

4.3 Data Recording and Retrieval

5. Simulation System Architecture

5.1 Simulation System Architecture

5.1.1 Math Model Execution Environment

The Simulation System Model Execution Environment provides a Real-time mode of model execution and a non-real-time mode (i.e., Remote Terminal Mode) of model execution.

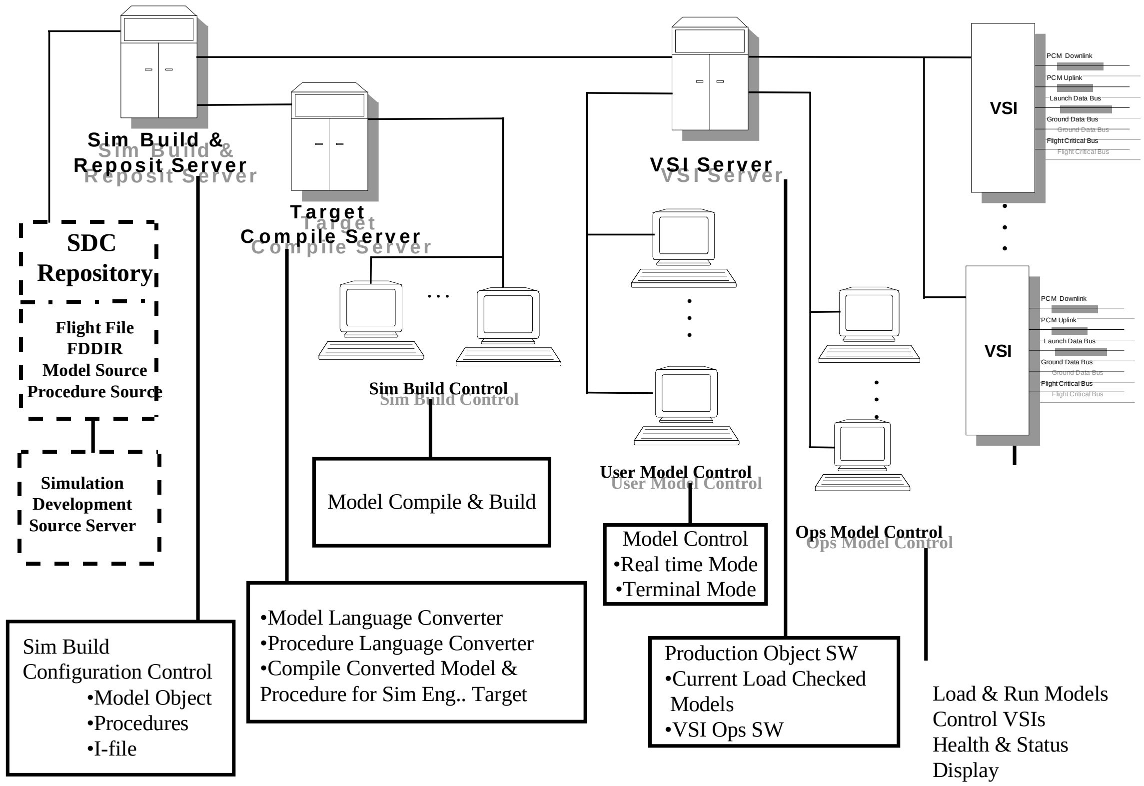
The Real-time Mode of Execution supports the execution and interfacing of models to the Real Time Processing System (RTPS) in order to provide check-out and validation of test procedures, Launch crew training, RTPS equipment checkout.

The Remote Terminal Mode of Execution supports the execution, testing, and checkout of models and procedures in the non-real-time mode of operation. This mode of execution supports the debug and preliminary check-out of simulation and application software. The Remote Terminal Mode of Execution provides user control and monitor of model and procedure execution from a remote terminal or workstation. In the Remote Terminal Mode all model stimuli and execution control must be entered from the terminal or by the use of a precompiled procedure invoked at the terminal.

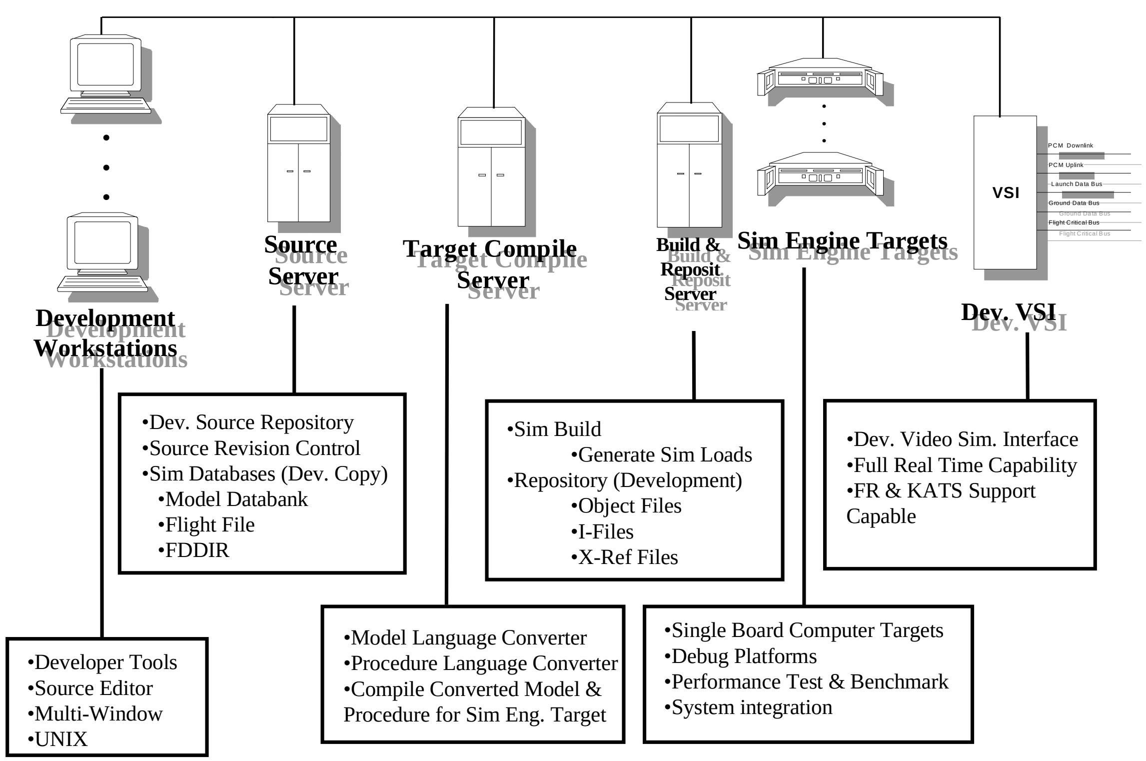
5.1.2 Real Time Simulation System Hardware Interface

The Simulation System provides real-time simulation support through the Video Simulation Interface (VSI). The VSI provides hardware signals that emulate the vehicle and ground system hardware interfaces to the RTPS Gateways. The Simulation System also includes an interface to the Real Time Critical Network (RTCN) that supports simulation without the use of the RTPS Gateways or VSI equipment. In this mode, the Simulation System emulates the RTPS Gateways interface with the RTCN.

5.1.3 Model Production Environment Architecture



5.1.4 Model Development Environment Architecture



5.2 Simulation Support Software

The Model Translator, Procedure Converter, and Model Build are Simulation Support Software components that support the Shuttle Ground Operations Simulator (SGOS) modeling language and Model Control Procedure language. These Simulation Support Software components acquire information from the Databank, the Flight Format File, and the Function Designator (FD) Directory (FDDIR) to provide interfaces with the RTPS.

5.2.1 Model Translator

The Model Translator takes an SGOS Math Model as input and, parses the SGOS modeling language syntax, generates a converted model source in C with model segments represented by C functions. During model translation, the translator accesses model databank information in order to verify correct model variable usage (i.e., discrete model variables operating on ON or OFF values versus analog model variables operating on integer or floating point values). A model variable cross reference and a list of model subsystems called by the model are generated by the Model Translator and utilized by Model Build.

5.2.2 Procedure Converter

The Procedure Converter takes a SGOS Model Control Procedure (MCP) as input and, parses the SGOS Model Procedure statement syntax, generates a C source procedure that can be compiled and executed by the Model Control Procedure Executive. During the procedure translation, the converter accesses model databank information in order to verify correct usage of databank names. In addition to the syntactical and variable checking, the Procedure Converter performs all error checking described in KSC-LPS-SGOS-5100, section 4. A list of procedures called by the source procedure generated by the Procedure Converter is. This is used by the Model Build process.

5.2.3 Model Build

Model Build takes as input the model variable cross reference and list of model subsystems generated during model translate. Model Build integrates a master model with model subsystems and provides model segment ranking. During Model Build, the Flight Format File and FD Directory (FDDIR) are accessed for PCM format and FD hardware information, respectively. Model Build generates tables that are required by the model executive (MEXEC), tables that are required to interface to the VSI, and an integrated executable for the Master Model for the hardware platform specified.

5.2.4 Databank Interface

The Model and FD databanks are accessed during either of a Model translate or Procedure Convert for type checking.

5.2.5 Flight File Interface

The Flight Format file on SDC is accessed during Model Build to obtain PCM format information.

5.2.6 FDDIR Interface

The FDDIR on SDC is accessed during Model Build to obtain FD hardware information.

ACRONYMS

	- A -
ACA	ADAS Command Application
ADAS	Advanced Data Acquisition System
AERO	Automated Electrical Retest Operations
ALS	Automated Logging System
AMRS	Automated Material Request System
ANSI	American National Standards Institute
AOS	Analog Overflow Signed
API	Application Program Interface
APU	Auxiliary Power Unit
ASM	Analog Serial Measurement
ATM	Asynchronous Transfer Mode
AWPS	Automated WAD Processing System
	- B -
B/U	Backup
BCD	Binary Coded Decimal
BFL	Block Funnel Logged
BIN	Business and Information Network
BTU	Bus Terminal Unit
	- C -
C&T	Communications and Tracking
CADS	Command and Data Simulator
CCC	Complex Control Center
CCMS	Checkout, Control, and Monitor System
CCP	Command and Control Processor
CCS	Complex Control Set
CDBFR	Common Data Buffer
CDT	Countdown Time
CEA	Common Equipment Area
CEU	Calibrated Engineering Units
CI	Configuration Item
CIE	Communications Interface Equipment
CITE	Cargo Integrated Test Equipment (Set)
CLCS	Checkout and Launch Control System
CM	Configuration Management
COLA	Collision Avoidance
COTS	Commercial Off the Shelf
CPDS	Computer Program Development Specifications
CPU	Central Processor Unit
CRMP	Command and Real-Time Monitor Position
CRT	Cathode Ray Tube
CS	Consolidated Systems
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CWLIS	Catenary Wire Lightning Instrumentation System
	- D -
DAP	Data analysis and Presentation
D/L	Downlink
DBSAFE	Data Bank Shuttle Automated Function Executive
DCN	Display and Control Network
DD	Data Dictionary
DDP	Data Distribution Processor

DFRC	Dryden Flight Research Center (Set)
DIO	Direct Input/Output
DLES	DPS LCC Expert System
DPS	Data Processing System
DSR	Display Synchronous Rate
	- E -
EDAMS	Engineering Data Access Management System
EDL	Engineering Development Laboratory
EIU	Engine Interface Unit
EU	Engineering Units
	- F -
FCAP	Facility Condition Assessment Program
FD	Function Designator
FDDI	Fiber Distributed Data Interface
FDID	Function Designator ID
FEP	Front End Processor
FOTE	Fiber Optic Terminal Equipment
FTP	File Transfer Protocol
FZ	Flow Zone
	- G -
GCP	Gateway Control Processor
GDB	Ground Data Bus
GMS	Ground Measurement System
GMT	Greenwich Mean Time
GOAL	Ground Operations Aerospace Language
GPC	General Purpose Computer
GSE	Ground Support Equipment
GUI	Graphical User Interface
G/W	Gateway
	- H -
HAZGAS	Hazardous Gas
HCI	Human Computer Interface
HGDS	Hazardous Gas Detection System
HIM	Hardware Interface Module
HMF	Hypergol Maintenance Facility (Set)
HOSC	Huntsville Operations Support Center
HUMS	Hydrogen Umbilical Mass Spectrometer
HWCI	Hardware Configuration Item
	- I -
IAPU	Improved Auxiliary Power Unit
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronic Engineers
I/O	Input/Output
IP	Internet Protocol
IPR	Interim Problem Report
IRIG-B	Inter-Range Instrumentation Group
ISO	International Standards Organization
IVT	Interface Verification Testing
IWCS	Integrated Work Control System
	- J -
JSC	Johnson Space Center
	- K -
KATE	Knowledge-Based Autonomous Test Engineer

KATS	Kennedy Avionics Test Set
Kb	Kilo-bit
Kbs	Kilo-bits per second
KB	Kilo-Byte
KBS	Kilo-Bytes Second
KEDS	Kennedy Electric Drawing System
KSC	Kennedy Space Center
KSDN	Kennedy Switched Data Network
	- L -
LACD	Local Acquisition, Command, and Display Subsystem
LAN	Local Area Network
LATMOS	Lightning and Transients Monitoring System
LDBM	Launch Data Bus Monitor
LCC	Launch Control Center
LCC	Launch Commit Criteria
LDB	Launch Data Bus
LO_2	Liquid Oxygen
LOC	Lines of Code
LON	LPS Operational Network
LH_2	Liquid Hydrogen
LPS	Launch Processing System
LRU	Line Replaceable Unit
LS	Launch Sequence
LSDN	LPS Software Development Network
LIVIS	Lightning Induced Voltage Instrumentation System
	- M -
Mb	Megabit
Mbs	Megabits per second
MB	Megabyte
MBs	Megabytes Second
MCP	Model Control Procedures
MDM	Multiplexer/Demultiplexer
ME	Main Engine
MER	Mission Evaluation Room
MET	Mission Elapsed Time
MFR	Multi-function Room
MFSC	Marshall Space Flight Center
MILA	Merritt Island Launch Area
MM	Mass Memory
MMU	Mass Memory Unit
MOIR	Mission Operations Integration Room
MPT	Mini-Peripheral Test (Set)
MSec	Millisecond
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
MTU	Mission Timing Unit
	- N -
NASA	National Aeronautics and Space Administration
NFS	Network File System
NSP	Network Signal Processor
	- O -
O&M	Operations and Maintenance
OCF	Orbiter Computational Facilities
OCR	Operations Control Room

OFI	Operational Flight Instrumentation
OIS	Operational Intercom System, also OI Standby (FEP)
OLDB	On Line Data Base
OLSA	Orbiter LPS Signal Adapter
OMI	Operations and Maintenance Instruction
OMRSD	Operations and Maintenance Requirements and Specification Documentation
OMS	Orbital Maneuvering System
OPF	Orbiter Processing Facility
ORT	Operational Readiness Test
OTV	Operational Television
OV	Orbiter Vehicle
	- P -
PAMS	Portable Aft Mass Spectrometer
PCC	Processing Control Center
PCL	Prerequisite Control Logic
PCM	Pulse Coded Modulation
PCMMU	Pulse Coded Modulation Master Unit
pF	Pico-Farad
PFP	Programmable Function Panel
PLC	Programmable Logic Controller
PMS	Permanent Measurements System
POCC	Payload Operations Control Center
POST	Power On Self Test
ppm	Parts Per Million
PRACA	Problem Reporting and Corrective Action
PSCNI	Program Support Communications Network Internet
	- Q -
	- R -
RADS	Remote Acquisition and Display Subsystem
RAID	Redundant Array of Inexpensive Disks
RCS	Reaction Control System
RCVS	Remote Controlled Video Switch??
RNET	Reconfiguration Network
RON	Restricted Operational Network
RSYS	Responsible System
RTCN	Real-Time Critical Network
RTPS	Real-Time Processing System
RTU	Remote Terminal Units
	- S -
SACS	Systems Software Avionics Command Support
SAIL	Shuttle Avionics Integration Lab (Set)
SBC	Single Board Computer
SCA	Sequence Control Assembly
SCAN	Shuttle Configuration Analysis Network
SCID	System Configuration Identifier
SCSI	Small Computer System Interface
SCT	System Configuration Table
SDC	Shuttle Data Center
SDE	Satellite Development Environment
SDS	Shuttle Data Stream
SDT	Shuttle Data Tape
SECAS	Shuttle Engineering Computer Application System
SGOS	Shuttle Ground Operations Simulator
SIM	Simulation System

SIMS	Still Image Management System
SL	Space Lab
SLOC	Source Lines of Code
SL-GMS	Sherrill-Lubinski-Graphical Modeling System
SLP	SSME Load Program
SLS	System Level Specification
SLWT	Super Light Weight Tank
SODN	Shuttle Operations Data Network
SONET	Synchronous Optical Network
SPDMS	Shuttle Processing Data Management System
SPF	Software Production Facility
SRB	Solid Rocket Booster
SSC	Stennis Space Center
SSME	Space Shuttle Main Engine
SSPF	Space Station Processing Facility
SSR	System Synchronous Rate
STM	Synchronous Transfer Mode
	- T -
TAB	TACAN Bearing (SDT Type)
TACAN	Tactical Air Navigation
TBD	To Be Defined
TCID	Test Control Identifier
TCMS	Test, Control and Monitor System
TCP	Transmission Control Protocol
TCS	Test Control Supervisor
TCS-1	Test Control Supervisor - Single Command
TCS-S	Test Control Supervisor - Sequence
TDM	Time Division Multiplexing
TEI	Test End Item
THDS	Time Homogenous Data Set
	- U -
UOPS	Utility Outage Processing System
USCA	Universal Signal Conditioning Amplifier
UTC	Universal Time Coordinated
	- V -
V&DA	Video and Data Assembly
VAB	Vehicle Assembly Building
VHMS	Vehicle Health Management System
VP	Vehicle Processing (Set)
VPF	Vertical Processing Facility
VSI	Video Simulation Interface
VTP	VME Telemetry Processor
	- W -
WAN	Wide Area Network
W/S	Workstation
	- X - Z -

GLOSSARY

Active — A term used to describe one of the subsystem computers in an Active/Standby pair that is currently performing its intended function.

Application — An application is a computer software program, or set of programs, written to perform a specific task (e.g., MS Word, MS Power Point, CCMS Ground Launch Sequencer). There are two different categories of applications in CLCS — Systems Applications and User Applications.

Application Software - The CLCS software used to monitor and control a specific set of end item systems. It consists of executable software that controls and monitors end items, and tables that define the interface to end item systems.

Application Services - System Software that is used to access the RTPS services required for User Application execution. Optionally, it may also be used by System Applications.

Backup — A subsystem that can be used as a replacement for one of the pair of Active/Standby subsystem computers in the event of a failure. Manual activity is required to bring the backup subsystem online (i.e., the backup system is not maintained in a "hot" standby role).

Business and Information Network — The Business and Information Network is a network of communications paths, computers, and computer programs that provide access to information that is external to the CLCS.

Bypass Command — A command which bypasses the End Item Manager.

Calibration - Calibration is the process of converting the digital representation of an analog measurement that has been acquired from a sensor to a floating point value that represents the calibrated engineering units of the measured quantity (e.g., pounds of pressure per square inch, temperature, speed, etc.). Sensors convert the measured quantities to an electrical voltage (an analog signal) which is subsequently converted to a digital quantity by an analog to digital converter (ADC). Sensors may introduce non-linearities into the measured quantities due to the physics of the measuring device. If the voltage output of the sensor has a linear relationship with the pressure measured by the sensor, engineering units can be calculated by the linear equation.

> Measurement Value = m * voltage + b, (where m is the slope of the line and b is the voltage value when the pressure is equal to zero.)

For non-linear signals a polynomial is required to convert the voltage value to the equivalent pressure value. An n^{th} order polynomial, of the form below, is used to make this conversion:

> Measurement Value = $\mathbf{a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x^1 + a_0}$, (where x represents the sensor voltage)

Calibrated Measurement Data — Analog data which has been converted to engineering units and calibrated by applying the calibration coefficients.

Cargo Integrated Test Equipment Set (CITE) — The Cargo Integrated Test Equipment Sets are located in the Space Station Processing Facility (SSPF) and Vertical Processing Facility (VPF) and are used to integrate and test payloads prior to them being loaded into the Orbiter.

CCMS Set — The set of equipment used to control and monitor end items using the CCMS. See table A-1 for a list of CCMS Sets.

CDBFR Counts — CDBFR counts are count values resident in the CDBFR. These counts are left justified within the CDBFR word. All CDBFR counts are assumed to be linear, bipolar, 2's complement, big-endian integer values. CDBFR count values can be up to 16-bits long. This terminology is new and is introduced to clarify the distinction between right justified processed counts and left justified CDBFR counts.

CDBFR Length — The CDBFR length of a measurement is the number of CDBFR bits required to represent the measurement in CDBFR counts. For example, the CDBFR length of an 8-bit unipolar GSE measurement is 9; 1 sign bit plus the 8 unipolar magnitude bits (remember that CDBFR counts are always assumed to be bipolar, thus a sign bit must be added to unipolar measurements by the FEP). The CDBFR length of a 10 bit bipolar PCM measurement is 10; the sign is already part of the raw data and does not have to be added. CDBFR lengths range from 2 to 32 bits. Since the CCMS CDBFR uses 16-bit words, any measurement with a CDBFR length > 16 requires two CDBFR locations

CDBFR Size — The CDBFR size of a measurement is the total number of CDBFR bits being used to hold the measurement. In CCMS, single word analogs (AM) reside in a single CDBFR location, thus the CDBFR size is 16. Multi-word analogs (AMDP) reside in two CDBFR locations, thus the CDBFR size is 32.

Calibrated Engineering Units (CEU) — See Calibrated Measurement Data.

Checkout and Launch Control System (CLCS) — The Checkout and Launch Control System is the replacement for LPS and is composed of a Real-Time Processing System, Shuttle Data Center, Simulation System, and a Business and Information Network.

CLCS Configurations - Alternate, scaleable, run-time equivalent RTPS configurations produced by a System Build.

1. <u>Full Configuration</u> - Supports full real-time RTPS performance and redundancy management. It contains no combined Subsystems.
2. <u>Limited Configuration</u> - Supports limited real-time RTPS performance and limited redundancy management. It contains DDP/CCP combined Subsystems.
3. <u>Minimal Configuration</u> - Does not support real-time RTPS performance or redundancy management. It contains DDP/CCP/HCI combined Subsystems. A minimal configuration may include internal modeling or a Simulation Gateway.

CLCS Environments - Each CLCS Environment has targeted purposes and supported configurations:

1. <u>Application Test-bed Environment</u> - Supports single user development and debug of all User and most System Applications. It normally uses a desktop Minimal Configuration and may employ an internal or external model.
2. <u>Integrated Development Environment</u> - Supports multi-user, fully integrated, debug, test and validation. It can be used for limited team training.
3. <u>Office Environment</u> - Supports monitor only User Display execution from the office areas. It permits the same User Displays used in the other environments to be used with conventional office remote PC hardware.
4. <u>Operational Environment</u> - Supports operational use of CLCS Sets. It is used to monitor and control end item hardware. It can be used for team training.
5. <u>Satellite Development Environment</u> - Supports multi-user development, debug and limited integration testing of System and Application Software

Command Panel — Command Panels are Input/Output (I/O) devices that provide information to and receive manual inputs from Console Position users.

Common Data Buffer (CDBFR) — A shared memory that contains the current status of all measurement data and subsystems in a CCMS set. The CDBFR also provides allows transmission of all messages in a CCMS set.

Common Equipment Area (CEA)— The Common Equipment Area is the back room of the OCRs and MFR and contains the Control Groups (CCP, DDP), Gateway Groups, and network equipment for the Launch Control Center Set.

Complex Control Set (CCS) — The Complex Control Set is located in the LCC at the Kennedy Space Center. It is used to monitor and control facilities equipment at KSC.

Computer Software Configuration Item (CSCI) — An aggregation of software and documentation that satisfies an end use function and is designated for separate configuration management by the acquirer. CSCIs are selected based on tradeoffs among software function, size, host or target computers, developer, support concept, plans for reuse, criticality, interface considerations, need to be separately documented and controlled, and other factors.

Configuration Item (CI) —An aggregation of hardware, software, or both that satisfies an end use function and is designated for separate configuration management by the acquirer.

Console — This term is used exclusively to describe the three-bay housing in the CCMS LPS system.

Console Back Row — This term is used exclusively to describe the CCMS set of MODCOMP computers, tables and PCGOAL workstations that reside behind the Consoles in the CCMS LPS system.

Console Position — The CLCS Console Position is a generic term used to refer to all the necessary workstations, hardware, peripherals and connectivity required (e.g., OIS-D, OTV, Safing, Applications Software Access, Business Systems Access, etc.) to support any the human interface to the RTPS.

Console Support Module — A CLCS Console Support Module provides the necessary table space, hardware and connectivity to allow additional users to comfortably support operations taking place at a CLCS Console Position.

Constraint Management — The capability to monitor Measurement FDs for a predetermined condition and notify personnel operating the Test Set and other software applications executing within the Test Set that the monitored data no longer meets the predetermined condition.

Control Group — A physical set of hardware consisting of Data Distribution Processors and Command Control Processors connected together along with the associated Real-Time Critical Network and Display and Control Network equipment.

Controlling — A term used to describe a software process that is currently actively monitoring and managing a set of hardware or software resources. The process normally has a second copy of itself that is executing in a separate CPU and monitoring the activities of the controlling process. On command, this second copy will assume control of the resources

Converted Counts — Converted counts are raw counts that have undergone subtype-dependent processing in the FEP to isolate and position count data within a data word. Examples of subtypes that require processing are Binary Coded Decimal (BCD), TACAN Bearing (TAB), BIT String Magnitude (ASM), and Halfword Overflow Signed (AOS). Other than the fact that converted counts have had some “conversion processing” performed, these counts are similar to raw counts, i.e., they are right justified, bipolar, 2’s complement, big-endian integer values. This terminology is new and is introduced to clarify the distinction between right justified raw counts and right justified processed counts.

Data Fusion — Data fusion is the process of combining measurement data and other CLCS system parameters into a higher level of information. The process for combining data may include algebraic and logical manipulation of data and conditional testing of system parameters or intermediate calculated values. The combined or “fused” data is a CLCS function designator.

Data Health — The combination of gateway status for each item of data and user defined health for the data item. Data health contains both health warning and failure information.

Data Status — The portion of data health that relates strictly to the status of the data item as determined by the gateway (e.g., a measurement who’s status indicates it isn’t being updated because it isn’t in the current PCM format).

Delivery – An incremental deployment of CLCS development consisting of facilities preparation, hardware, and software.

Delivery Theme – The primary driving force for determining the capabilities to be deployed in a CLCS delivery.

Derived Measurement — A Derived Measurement is one whose value is created within the RTPS.

Direct Command — A command to the End Item Manager to disable closed loop control of a function and allow issuance of a command to the end item itself.

Display Attribute – Information relating to a piece of data being displayed (e.g., color).

Display Synchronous Rate (DSR) — The rate at which data is distributed for display.

Dryden Flight Research Center (DFRC) Set — The Dryden Flight Research Center Set, located at Edwards, California, is used to monitor and control the Orbiter during landing operations.

Endian Architecture — When several bytes are required to store data, computers differ in which byte of storage they consider to be the “first”. In “right-to-left” or “little-endian” architecture, which include the Intel 80x86 and Pentium microprocessor, the address of a 32-bit integer is also the address of the low-order byte of the integer. In “left-to-right” or “big-endian” architectures, which include the Motorola 680x0 microprocessor family, the address of a 32-bit integer is the address of the high-order byte of the integer. Consider the number 1234 that is stored in addresses a through a+3:

Big-endian, left-to-right	01	02	03	04	The number is stored left to right
	a	a+1	a+2	a+3	
Little-endian, Right-to-left	04	03	02	01	The number is stored right to left
	a	a+1	a+2	a+3	

End Item — The set of actuators, transducers, software and equipment that are controlled by CLCS. This term is used to indicate both the lowest level of end item (e.g., a valve) and the collection of end items that constitute a larger end item (e.g., APU, Orbiter, Payload, Facilities Equipment).

End Item Manager — The software used to implement closed loop control of End Items within CLCS.

Engineering Units — Engineering units are the representation of the measurement in the units understood by the user (e.g., volts, PSI, feet per second, etc.). Floating point numbers are used to represent engineering units. The engineering unit range depends on the type of hardware transducer.

Flight Vehicle — The term used to refer to the Orbiter, External Tank and Solid Rocket Boosters; this term does not also refer to the Payload.

Flow Zone — A collection of console positions configured to support a particular test. Within an OCR, one or more flow zones can exist and will fluctuate according to the test configurations.

Focal Thread — Each CLCS delivery will have one to two focal threads that are used to integrate work from multiple system threads. Taken together, the focal threads provide the delivery theme capability and normally focus on the user.

Forward Link Mode — A mode in which the RTPS PCM Uplink can operate, in which data is transmitted at 128 Kbs to the Orbiter Communications Interface Equipment (CIE) via the ground Communications and Tracking (C&T) station.

Fractional Counts — Fractional counts are a floating point “interpretation” of CDBFR counts that assumes an implied binary point exists to the right of the sign bit. Mathematically, then, fractional counts range from -0.5 to +0.5. CCMS console software converts CDBFR counts to fractional counts prior to scaling.

Function Designator (FD) — Data processed by the CLCS which has been acquired from the End Item under test, calculated by Data Fusion, determined by System Software, derived by a user test application or the last commanded value of an output command. Measurement FDs consist of the following specific FDs:

1. End Item FDs
2. Data Fusion FDs
3. System Status FDs
4. Pseudo FDs
5. Commanded Status FDs

Function Designator Data — The set of data relating to a function designator. The data consists of FD Value (i.e., Converted Counts, Calibrated Engineering Units), FD Health, FD Time of occurrence, and Display Attributes.

Gateway Group — The collection of Gateways which are physically linked to a particular physical area (e.g., Pad A, OPF 1, etc.).

Hardware Configuration Item (HWCI) — An aggregation of hardware that satisfies an end use function and is designated for separate configuration management by the acquirer.

Hot Standby — An on-line resource ready to be configured to serve as a replacement for another resource.

Hypergol Maintenance Facility (HMF) Set — The Hypergol Maintenance Facility Set is located in the industrial area at the Kennedy Space Center. The HMF Set is used to monitor and control Hypergolic equipment from the Orbiter that is being reconditioned and refurbished for reuse.

Intrusive — Any process or event which will significantly alter the normal configuration or operation of an on-line resource.

Kennedy Avionics Test System (KATS) Set — The Kennedy Avionics Test System Set is located in the VAB at the Kennedy Space Center and is used to provide limited avionics test and simulation capability without utilizing the larger Shuttle Avionics and Integration Lab at the Johnson Space Center. KATS also supports procedure development and, in a limited capacity, pre-installation flight element LRU checkout.

Launch Control Center (LCC) Set — The Launch Control Center Set consists of all of the RTPS equipment located in the OCRs, CEA, MFR of the LCC at the Kennedy Space Center. This equipment may be configured into subsets or supersets to accommodate tests requiring varying RTPS resources to support appropriate End Items.

Launch Data Bus Monitor (LDBM) — A collection of equipment used to record all message traffic on the LDB and GSE data busses.

Linearization — Signal non-linearities are introduced into analog measurements by signal conditioners and transducer non-linearities. The CCMS system assumes that the analog data, which is stored and processed from the CDBFR, exhibits a linear relationship between the signal (in counts) as recorded in the CDBFR and the actual signal fluctuation as measured by the transducer. A method of accounting for this non-linearity was implemented in LPS. The capability was provided to associate line segment approximations of the calibration data curve with an analog measurement. The linearization curve is approximated by the use of one or four line segments. Each line segment has a K1 (slope) value and a K2 (Y-intercept) value which constitutes the standard equation for a line.

Local Operations — Local Operations are those that are performed in the vicinity of the actual hardware.

Logged Data — Data that is sent to the recording system to be recorded.

Mini-Peripheral Test (MPT) Set — A KSC CCMS test set used to maintain CCMS hot spare equipment.

Multi-Function Room (MFR) — The Multi-Function Room is one of the rooms which was formerly a KSC Launch Control Center (LCC) Firing Room. The MFR contains the Flow Zone and Shared I/O Area used for Flight Vehicle, Payload and GSE Processing.

Non-Controlling — A term used to describe a software process that is designated to replace the Controlling process in the event of a failure. The Non-Controlling process monitors the system configuration but does not actively manage the configuration unless the Controlling process fails.

Non-Intrusive — Any process or event which will not significantly alter the normal configuration or operation of an on-line resource.

Non-Operational — The state of a resource known to not be ready to support an activity.

Off-line — A term used to signify a physical resource is neither ready to participate nor is participating within a Set.

On-line — A term used to signify a physical resource is ready to participate or is actively participating within a Set.

Operational — The state of a configured, initialized resource that is ready to support.

Operational Readiness Test (ORT) — A set of software used to verify that a subsystem's hardware and software is ready to support testing.

Operations Control Room (OCR) — Operations Control Rooms are rooms which were formerly KSC Launch Control Center (LCC) Firing Rooms. The OCR contains the Flow Zone and Shared I/O Area used for Flight Vehicle, Payload and GSE processing.

Override — An override is a keyboard command to Prerequisite Control Logic (PCL) requiring PCL to issue a command that has failed its prerequisite checks.

Platform Load — The combination of the tailored Operating System (OS) and COTS software which may be loaded onto a CLCS Subsystem HW.

Prerequisite Sequence — A prerequisite sequence may be defined for every end item command. When a command with a prerequisite sequence is issued from the keyboard or an application program the prerequisite sequence is executed. If the prerequisites are passed, the command is issued; otherwise, it is rejected. A rejected command may be overridden by the operator.

Processed Counts — In CCMS, processed counts are raw counts that have undergone subtype-dependent processing in the FEP (i.e., converted counts). Processed counts have been linearized by the FEP using line segment data from its' tables, if the measurement requires it, and are bipolar, 2's complement, big-endian integer values. The CCMS usage of the term may imply right-justified counts or left-justified counts (see CDBFR counts).

Pseudo Function Designator — A pseudo FD is a FD that is updated by applications within the RTPS. Pseudo FDs have many of the characteristics of measurement FDs (e.g., they are defined in the CLCS Data Bank, have On-line Data Bank entries, may be used by Data Fusion, etc.).

Raw Counts — Raw counts are the count values as received from the hardware. These counts are right justified. Raw counts are represented by an Integer value whose range depends on the measurement length. The normal count range for an 8-bit unipolar GSE measurement is +3 to +253. A 10 bit PCM bipolar measurement ranges from -512 to +511.

Real-Time Processing System (RTPS) — The Real-Time Processing System consists of Gateway Groups, Control Groups and Flow Zones. This equipment performs the real-time processing for CLCS.

Reduced Capability Mode — A mode of the CLCS in which all normal functionality defined for the Test Set is not present. This mode may be entered in a number of ways.

Redundant Pair — A term used to represent the association of an Active/Standby resource.

Responsible System — The set of users and applications responsible for controlling and monitoring a given end item system.

RTPS Set — The collection of equipment in a physical facility that is interconnected and dedicated to performing a major function or task (e.g., CITE is used for Payload Integration and Checkout). The equipment contained in a RTPS Set consists of the equipment contained in Flow Zones, Control Groups, Gateway Groups, and associated Networks (RTCN, DCN, & RON). Table A-1 lists the CCMS and RTPS Sets.

Set Name	CCMS Set	RTPS Set
Launch Control Center (LCC) Set	N	Y
Cargo Integrated Test Equipment (CITE) Set	Y	Y
Complex Control (CCS) Set	Y	**TBD**
Firing Room Number 1	Y	N
Firing Room Number 2 (Multiple Sets)	Y	N
Firing Room Number 3	Y	N
Hypergol Maintenance Facility (HMF) Set	Y	Y
Dryden Flight Research Center (DFRC) Set	Y	Y

Set Name	CCMS Set	RTPS Set
Kennedy Avionics Test System (KATS) Set	N	**TBD**
Mini-Peripheral Test (MPT) Set	Y	N
Processing Control Center	Y	N
Shuttle Avionics Integration Lab (SAIL) Set	Y	Y
Development Sets		
Integrated Development Environment (IDE) Set	N	Y
Satellite Development Environment (SDE) Set	N	Y

Table A-1 - CCMS and RTPS Sets

Safing System — A collection of hardware and software that provides a means, with minimal or no RTPS availability, to place a facility or Flight Vehicle into a known and safe state.

Scaleable Architecture — The RTPS Scaleable Architecture provides the capability to build the Minimal, Limited, and Full RTPS run time equivalent configurations.

Shareable Library — A collection of object code which may be linked with other object code at run time

Shared I/O Area — The area in an Operations Control Room (OCR), Multi-Function Room (MFR), or Specialized Processing Site containing printers, plotters, and other devices that are shared by multiple users in the Test Set.

Set Point — The value of a function in a closed loop application that the End Item Manager is maintaining.

Shuttle Avionics Integration Lab (SAIL) Set — The Shuttle Avionics Integration Lab Set is located at the Johnson Space Center. The SAIL Set is used to verify CLCS Software in the avionics integration environment at SAIL.

Shuttle Data Center (SDC) — The collection of hardware and software that provides the capability to build system and test software and monitor data via Constraint Management.

Shuttle Data Stream (SDS) — The Shuttle Data Stream is used to provide data to users outside of the controlled environment for the purpose of monitoring operations. This data can be certified or uncertified based on location, purpose of use and configuration control. There is a CLCS SDS as well as a CCMS SDS.

Simulation System — The portion of the CLCS that provides support for the testing and validation of CLCS equipment; checkout and validation of procedures used in ground testing and launch operations; training of CLCS console operators; and launch team training

Software Release — A collection of files, together with their version number, that make up a software build.

Specialized Processing Site — An RTPS set used in a location other than the KSC LCC to perform checkout, control and monitoring of local end items.

Standby — A term used to identify which one of the computers in an Active/Standby pair is designated to replace the Active in the event of a failure. The Standby subsystem is maintained in a “hot” standby state (i.e., it can immediately assume the role of the failed Active).

Station Set — A Station Set is a facility and all of its equipment (e.g., OV 104, Pad A, High Bay 1, OPF 1).

Subsystem — The collection of hardware and software that is combined to perform a specific set of functions (e.g. GSE Gateway, CCP, DDP).

Subsystem Hardware — The Hardware Configuration Items (HWCI) that are required to perform the functions of a subsystem.

Subsystem Load — The Computer Software Configuration Items (CSCI) that are combined to perform the functions of a subsystem.

System Application — The System Software that implements the subsystem functionality of an RTPS System. System Applications are normally transient applications that are either executed when a subsystem enters an operational mode or when a specific service performed by them is requested. System Applications utilize either System Services or Application Services to assist in accomplishing their task.

System Build — The build process required to create the System Software load of RTPS subsystems (i.e. Subsystem Loads). The build product is a System Configuration ID (SCID) which supports all CLCS Configurations. It also identifies the Operating System Load versions required for each Subsystem

System Configuration Identifier (SCID) — System Configuration Identifier is a CLCS term used in two different instances:

1. The label used to identify the System Build products constructed for a CLCS release
2. The collection of System Build products used in the RTPS subsystem computers.

System Services — The System Software that supplies features necessary to implement RTPS subsystem. It provides a portable layer for vendor specific COTS operating systems and COTS product services. It is used by System Applications and Applications Services.

System Software — The CLCS software that provides the development, build, and execution framework for operating a CLCS set and executing Application Software. System Software is incapable of monitoring or controlling end item systems without the use of Application Software

System Software Build — The process of creating software that is ready for loading into the RTPS subsystem computing equipment.

System Synchronous Rate (SSR) — DSR and user and system applications in the real time portion of the system.

System Tables - System Software that is non-executable and is loaded into various subsystems within the RTPS in order to permit proper execution of System Software for operation of an RTPS set.

System Thread — The set of software in a CLCS delivery that provides a system-wide capability.

Test Article — Test Article is a little used term to specifically define a system and may be composed of multiple End Items (e.g., the APU System is a Test Article).

Test Build — The build process required to create the Application Software (User Applications and User Tables), load files required to support RTPS monitoring and control of end item systems. The build product is a Test Configuration ID (TCID) which supports one or more System Builds.

Test Configuration Identifier (TCID) — Test Configuration Identifier is a CLCS term used in two different instances:

1. The label used to identify the Test Build products constructed for a CLCS release
2. The collection of Test Build products that define the set of end items to be tested.

Test Set — A group of CLCS RTPS Subsystems and networks configured for a specific test. Typically a Test Set will include one or more Gateways within a Gateway Group, some portion of an RTCN's bandwidth, one or more DDPs and CCPs within one or more Control Groups, some portion of a DCN's bandwidth, and one or more Console Positions. This entire set of equipment would constitute a Test Set. For example, a Launch configured Test Set would include all the Gateways in that Pad's Gateway Group, RTCN, two Control Groups with several DDPs and a larger number of CCPs utilized, the DCN within that OCR, and a Flow Zone consisting of all the Console Equipment in that OCR.

Thread — See System Thread.

Time Homogeneous Data Set (THDS) — A group of measurement data or measurements which must be viewed and updated as a group.

User — The term user is used in this document to mean a human being that is using CLCS equipment (hardware and software) to perform a given task. There are many users of the CLCS system

1. Operations user — one who uses the CLCS to process and or control end items.

2. Build user — one who uses the CLCS to build, system or test, software products to be used in the CLCS.
3. O&M user — one who operates and maintains the CLCS equipment.

User Applications — Application Software written by users. Most user applications are developed to run on the RTPS and are used to control and monitor End Items..

User Application Registration - The method used to dynamically connect User Applications to other User Applications and Function Designators. It ensures that all required services exist and are used properly, performs registration at User Application program startup to ensure compatibility and immediate availability, verifies external references. Additionally, it is used during the Test Build phase to verify and construct tables for running subsequent Inverted Index queries to assess impacts of future User Application changes.

User Tables - Application Software that is non-executable and is loaded into various subsystems within the RTPS in order to permit proper execution of User Applications and to define the interface with end item Function Designators.

Workstation — A desktop computer. In a CLCS OCR a workstation is a desktop computer utilized in a CLCS Console Position or Console Support Module. There are two kinds of workstations in CLCS:

1. Workstations for executing CLCS applications and viewing CLCS displays. This workstation could have more than one monitor.
2. Workstations for business systems connectivity to access on-line documentation and historical log books, generate deviation paperwork (e.g., IPRs, PRACA, Deviation Records) and to provide connectivity to center-wide and world-wide information.